MATHEMATICAL FORMULATION OF JOB SHOP SCHEDULING

BY

FRITZ NENTWICH

I. Staatspruefung, University of Technology,
Vienna, Austria, 1963

Dipl. Ing., University of Technology,
Vienna, Austria, 1966

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1968

Approved by:

J. E. Grosh
Major Professor

TABLE OF CONTENTS

1. INTRODUCTION

1.1 THE PROBLEM

This paper describes the vital problems of production planning for lot type production, which have been developed in the last few years. The production model we assume, consists of a sequence of operations that manufacture and assemble several components to one assembly on a certain number of facilities.

The problem to plan the production schedule is discussed in two parts:

Part I: Requirement generation

1. part explosion
2. demand and requirement,

Part II: Scheduling

3. lot size, and
4. scheduling.

The total scheduling problem shall be explained in this introductory pages step by step. The initial data are given with the sales forecast, which dictates the type of part to be produced, the quantities of these parts and the time when they have to be completed. How do we get a feasible and optimal schedule that meets all requirements? It is impossible to solve the problem in one procedure. Hence, the problem is split into several smaller subproblems.

The part structure descirbed in the bill of material, indicates all ingredients for the assembly. This information is used to generate the total number of basic elements which have to be manufactured to obtain one assembly. The explosion problem uses matrix algebra to describe part structure and to determine the total number of components necessary to assemble one unit of a specific commodity. The demand problem computes for the given sales forecast of assemblies and components the total required amount of components to fulfill the market need. The sales forecast given for several future periods, causes an interweaving of all these demands. The current production and the planning for the next periods have to be properly connected. All the quantities calculated in the demand problem are 'gross requirements'. They must be reduced or netted by the existing parts from the inventory and work in process to get the net requirements. These net requirements have to be produced. Here, the problem of every lot type production arises, the right lot size and the scheduling of these. The lot size problem shows how to evaluate an economical lot size, considering all possible circumstances of required quantities, production facilities, inventory, and machine preparation.

All we have described at this point is just preparation for the scheduling. The schedule is generated from start and completion times, and the total requirement quantities grouped in the optimal lot size. The production schedule is

filled into a wide time interval or "band," within which the production must be performed. The overall schedules are determined with the aid of "manufacturing bands," in which the foreman does the detail scheduling with the help of a loading rule for the facilities.

1.2 PRODUCTION PLANNING AND CONTROL

The general function of production planning and control in the total business system is described to give an overall view. These pages should indicate the connection to other problems of the production system, and to show where and how these problems arise in the total system. The different types of production are discussed to compare them and to point out where our technique can be applied.

The schematic block diagram in Figure 1.1 illustrates the interaction of the most important departments of a manufacturing firm. The lines connecting the different blocks show the flow of instruction (full arrows), feedback (dashed arrows) and material (open arrows).

1. Sales organization

The sales organization supplies management with information about the market situation, specific customer needs, and the sales forecast prepared from a market study.

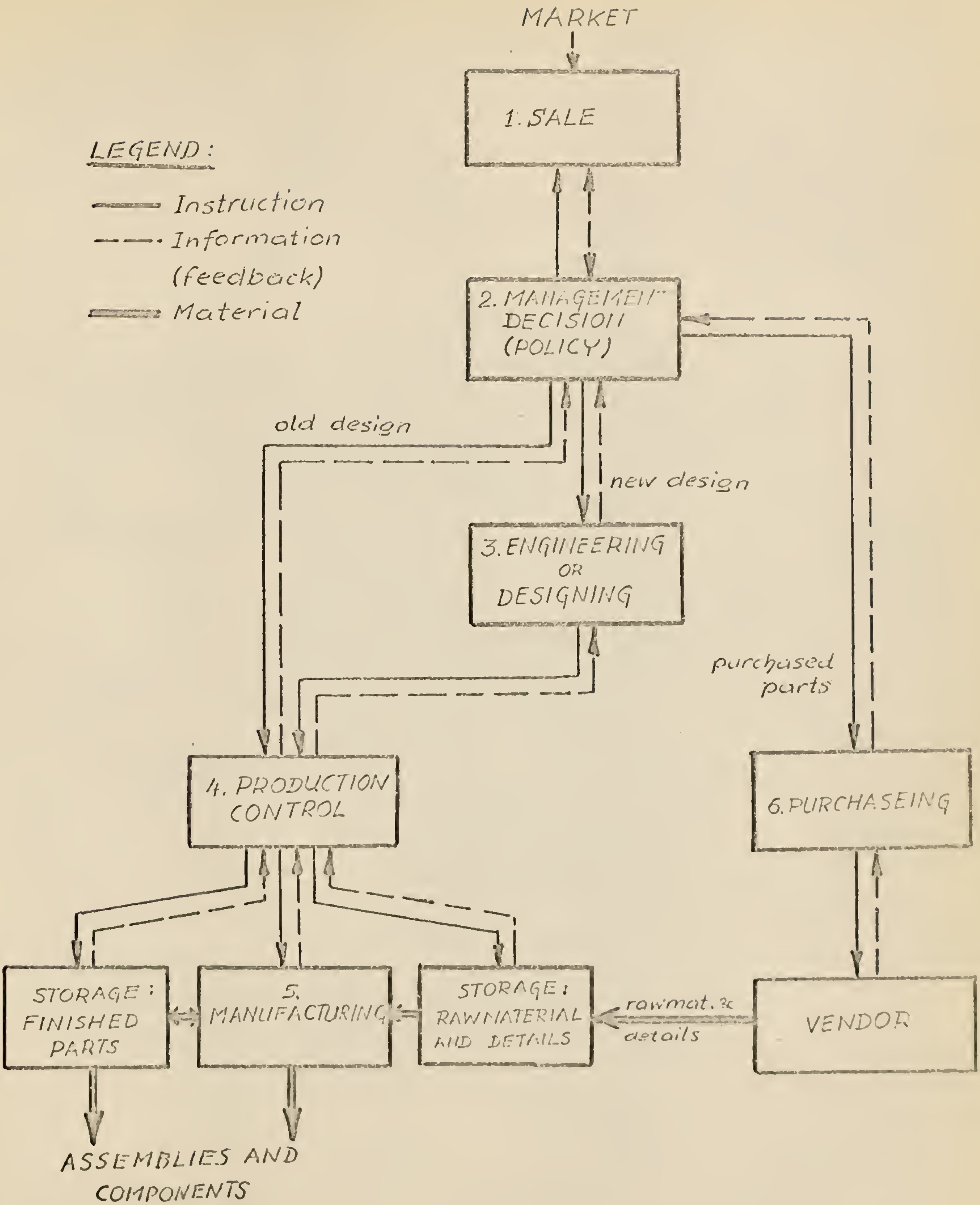


Figure 1.1: Instruction-, feedback- and material-flow in a business system.

2. Management decision function

The management decision function is to control the overall direction of the total business system. All policy statements originate here. To accomplish this goal, the decision group requires information from all the main departments.

3. Engineering or design group

This department designs and develops plans of the articles to be manufactured. The word design includes the preparation of the engineering drawings, the major directions for the operations, fixing of quality level and, with that, the required tolerance limits.

4. Production planning and control

Manufacturing the required quantity of a product in the desired quality at the correct time with the most economical method is the main function of this department.

5. Manufacturing

The production facilities (machines and equipment), the manpower and the process methods, which perform the actual task necessary to refine the material into the required form are the manufacturing department.

6. Purchasing

The purchasing department controls the vendor or sub-contractors. Detail parts and raw-material which are needed to accomplish the companies aim are selected, ordered, checked and controlled by the purchasing department.

1.3 TYPE OF PRODUCTION [9]*

The type of production depends upon the quantity of the finished product and the regularity of manufacture. According to the manufactured quantities involved, we distinguish three main groups:

a) Job production,

b) Lot production, and

c) Continuous production.

1.31 Job production

Job production is the manufacturing of the products to meet specific customer requirements of special orders. The quantity involved is relatively small. This production is normally concerned with special projects, models, prototypes, etc. Examples are: turbo-generators, large engines, shipbuilding, etc.

1.32 Lot production

In the lot or batch production a number of identical articles is produced, either to meet a specific custom order,

*The number in square brackets cites the reference in Chapter 7.

or to satisfy continuous demand. When production of the lot is terminated, the plant is available for production of similar products. Similar to job production, the policies regarding tooling and other aids depend on the quantities involved. If the order is executed only once there is less justification for providing elaborate production aids than when the order is repeated.

In lot production three types can be mentioned, according to the regularity of the production:

1. A lot is produced only once.
2. A lot is produced repeatedly at irregular intervals, when the demand occurs.
3. A lot is produced periodically at known intervals, to satisfy a continuous demand.

Here, planning and control are simplified as quantities increase and as manufacturing becomes more regular. In the lot production two basic problems arise: the correct size of the lot and the scheduling of the production. The attack upon these problems depends on whether production is governed by external orders only or by internal consumption or both. For external orders, the lot size is mainly determined by the customer to suit <u>his</u> specific circumstances. The plan is mainly concerned to meet the set delivery dates. For internal consumption, the plant produces to stock. Both the lot size as well as the scheduling problems are matters for internal management decisions.

The optimal lot size considers setup costs, which are involved before each production run, and the carrying costs created when the finished product is held in stock. The lot size determines the length of the production run and affects both the production schedule and lot size consideration of other products. These problems are further discussed in Chapter 4. Lot production is a very common feature in the industry. Examples are machine tool work, chemical processes, etc.

1.34 Continuous production

Continuous production is a specialized manufacturing of identical items on which the facility is fully engaged. It is associated with large quantities and a high rate of demand. Here, full advantage is taken of repetitive operations in the design of production aids such as special tools, transfer machines, inspection devices, fixtures, etc. There are many cases where plants are not confined to one particular type of production. Even very large companies involved in manufacturing in continuous production often use lot production for most of the components required on the assembly line. Production planning and control in such plants become rather substantial owing to the different types of production.

1.4 INTERACTION OF PROBLEMS

Figure 1.2 shows schematically the interaction and connection of the problems which are treated in this paper.

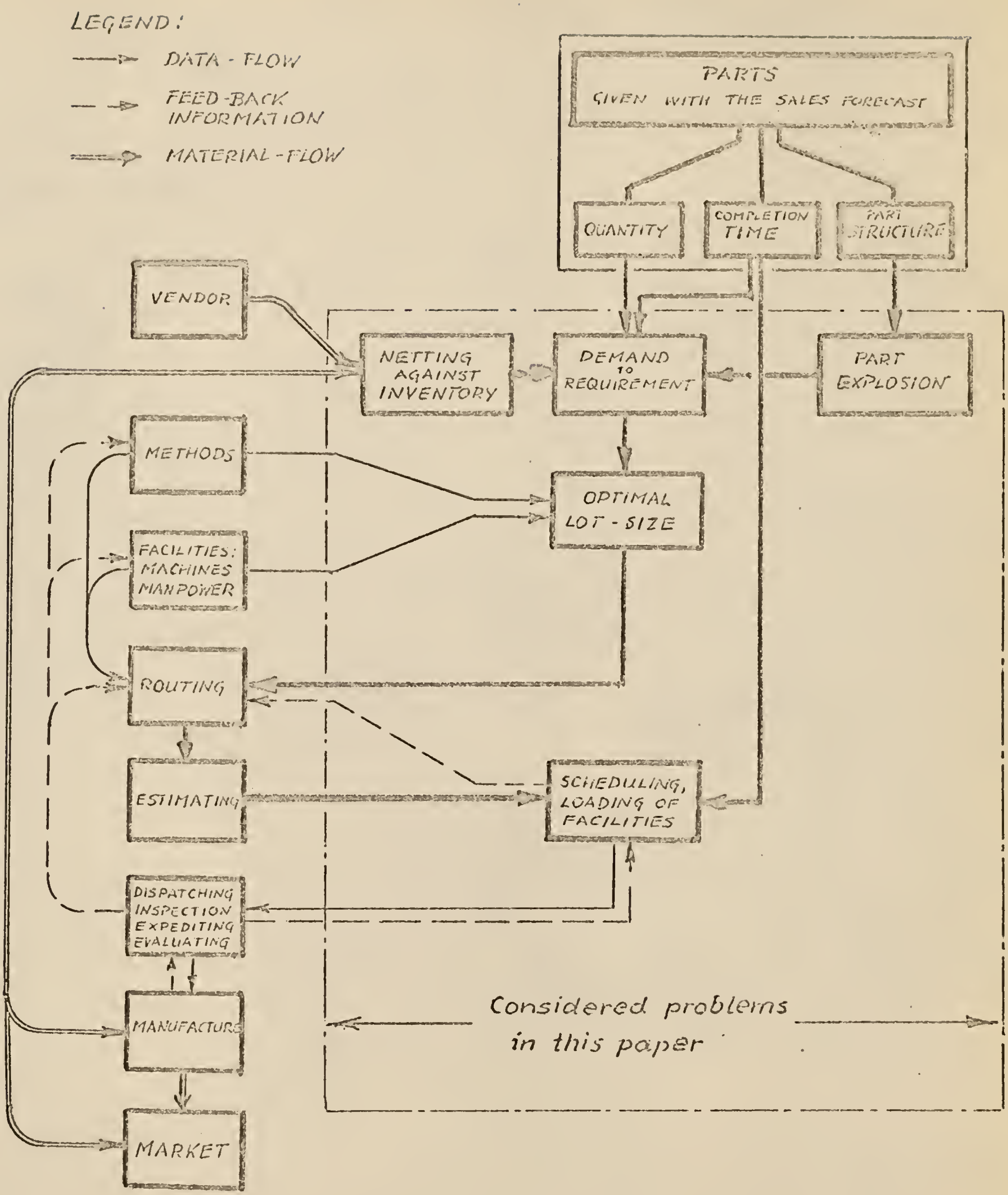


Figure 1.2: Interaction of the problems.

2. PART EXPLOSION

2.1 INTRODUCTION

A complex product made of several components must be processed in a series of steps. In each of these steps various parts, in integral amounts, are combined to create a new part--an assembly. In every one of the steps a certain number of identical articles, the lot or batch, is processed with the same production operations. We will develop a technique to describe the component-assembly-relation, graphically as well as mathematically. Matrix algebra is used to represent this relationship and to calculate the total number of components which are integrated into the manufactured part.

Example: [4]. The process of determining the number of parts required to build a planned lot of a product is called "requirements generation"; see chapter 3. For example, to manufacture one jeweled box, the parts shown in Figure 2.1 are needed for assembly. To build 300 such boxes, obviously each quantity listed times 300 are needed. If we purchase all parts for the box except the jeweled hinges, which are assembled from simpler parts, a part list for each hinge assembly must be generated (see Figure 2.2). To determine the number of detail parts required in 300 boxes, we have to multiply 300 boxes by 2 hinges per box times the quantity per hinge. For example each box requires a total of 28 jewels, 16 jewels for the box itself, and 6 jewels for each of the two hinges.

Part	Part No.	Quantity
box lid	1	1
box base	2	1
box side	3	4
hinge	4	2
screw	5	24
jewel	6	16

Figure 2.1. Part list or bill-of-material for one jeweled box, part no. 9.

Part	Part No.	Quantity
hinge leg	7	2
pin	8	1
jewel	6	6

Figure 2.2. Part list or bill-of-material for one hinge, part no. 4.

2.2 DEFINITION OF BASIC TERMS

In this section several basic terms will be defined. Additional terms and concepts will be explained as they occur.

2.21 Operations

An operation is a procedure that uses a facility to alter the physical, chemical or location state of the part being manufactured.

2.22 Production facility

A production facility is a man or machine or man-machine combination which performs operations. Facilities are defined as mutually exclusive and add up to the total of available facilities. Facilities are also defined, jointly with the operations, so that each operation requires exactly one facility for its performance.

2.23 Operation sheet

The operation sheet is a list of manufacturing operations that must be performed in the stated sequence on the specified facilities to convert a certain amount of a particular raw material, detail parts, and assemblies into a finished part.

2.24 Part

Final assembly or end-item. A part not assembled to another part, i.e., not a component of any other assembly in the manufacturing system, is defined as a final assembly (or as an end-item). Such a part represents the output of the manufacturing system and is shipped to a customer or to another department of the company, which is outside of our consideration. A final assembly does not appear as a consumed sub-part (or component) on any bill of material in the system.

Main-assembly. We call an assembly, consumed directly by the final assembly a main-assembly. The main-assembly is a main component of the final product. The components of the main-assembly are sub-assemblies of different levels, detail-parts, and raw material.

Sub-assembly. A sub-assembly, integrated from a certain number of parts, is a component of a main-assembly or another sub-assembly with a lower level number. It may be directly or indirectly consumed by the main-assembly.

Detail-part. A detail part is a 'discrete' component of any type of assembly, and requires no part to assemble itself. Detail parts are directly delivered from other companies or other departments. The parts are the 'input' of the manufacturing system. In opposition to raw material, the detail parts are countable items and are consumed in integer amounts. We include in the word "detail part" every part which is not assembled in our total system. Hence, every purchased part, even assembly, will be defined as a detail for our system.

Raw material. Raw materials are components of any type of assembly, and have the same property as the detail parts with one exception: raw material is a measurable item and is consumed in continuous quantities, for instance, length, area, volume, etc.

Part. The term part (or commodity) is the name applied to an item during and after manufacturing. The part will be said

to be "in process" until the last operation is completed. The term part is the general term, integrating final-assembly, main-assembly, sub-assembly, detail and raw-material.

Assembly. A part built up from a certain number of other parts is called an assembly. We say, a certain number of components are "consumed" to generate an assembly. The existence of the components for the assembly is, of course, not sufficient. To obtain the assembly an "assembly operation" is necessary.

2.25 Direct and indirect consumption

We say, a certain number of sub-parts are "consumed" to generate an assembly. A part "i" is "directly" consumed by assembly j, if it is directly required in the assembly operation. For the "direct" consumption of part i in assembly j, part i is a positive quantity in the bill of material of assembly j. Part i is "indirectly" consumed by assembly j, if part i is consumed (directly or indirectly) by a directly consumed sub-assembly of part j.

2.26 Bill of material

The bill of material is a list of parts and their quantities which are directly consumed in the manufacture of a given part. The bill of material carries three important pieces of information:

1) the name of the assembly j being made,
2) the name of the directly consumed components i, and
3) the quantity of the directly consumed components i.

2.27 Bill of consumption

Analogous to the bill of material, the "bill of consumption" is the list of parts and their quantities, where the specific part is consumed or used. The bill of consumption contains three pieces of information:

1) the name of the component i being used or consumed in other parts,
2) the name of the assemblies j, which consume or require this component i, and
3) the quantities of the direct consumption of component i in assemblies j.

We shall not use this definition further, it is mentioned here for completeness. This type of information is not normally maintained by manual systems.

2.28 Level order

The hierarchy of parts (i.e. final-assembly, main-assembly, assembly, and detail-part) leads to the expression "level". Parts, that require the same maximum number of assembly steps to appear in the end-item, are parts of the same level. We assign to the final-assembly the level number "1", as the highest

level; to the main-assembly the level number "2", as the next lower level; and so forth, till we reach the detail-part with the highest level number δ, as the lowest level. See Figure 2.13 in section 2.84.

FINAL ASSEMBLY	level number	1	HIGHEST LEVEL
MAIN ASSEMBLIES		2	
SUB-ASSEMBLIES		3	
SUB-ASSEMBLIES		4	
SUB-ASSEMBLIES		5	
DETAIL PARTS AND RAW MATERIALS		δ	LOWEST LEVEL

Figure 2.2a. Level order.

2.3 PRESENTATION OF THE PART STRUCTURE

2.31 Bill of material

The configuration of a part is described with the bill of material or part list. See the explanation in section 2.26 and the examples in Figure 2.1 and 2.2.

2.32 Part structure diagram

The configuration of an assembly can be presented clearly with a part structure diagram.

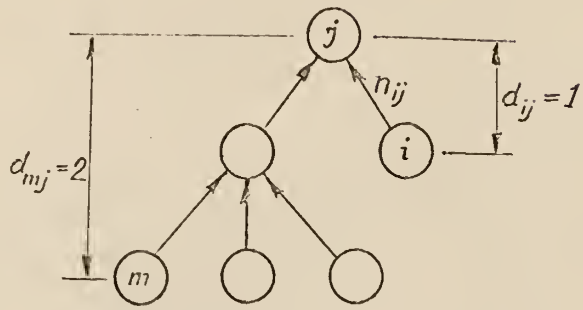


Figure 2.3. Part structure diagram.

Each PART is shown by a small circle or node, which includes the part number. The DIRECT CONSUMPTION of one part by another is represented by a line connecting the two circles. The arrow points to the consuming assembly. The number n_{ij} beside or on each line represents the QUANTITY of the entering part i, which is necessary to produce one unit of the assembly, j. If circle j lies above circle i in the part structure diagram and x arrows connect these two circles, we say part j is at a DISTANCE $d_{ij} = x$ from part i. Moreover, any circle is at distance zero from itself. The direct consumption of one part into another is at distance one. Concluding, we show a legend of the symbols used, in Figure 2.4.

Description	Graphical symbol	
part	circle	i
part number	number in the circle	i
direct consumption	arrow, pointing to consuming assembly	$i \rightarrow j$
consumed quantity per assembly	number next to arrow	$i \xrightarrow{n_{ij}} j$

Figure 2.4. Legend of the symbols of the part structure diagram.

Example. The part structure diagram of the jeweled box assembly identified with the number 9, is drawn in Figure 2.5. The final assembly 9 is shown in the top row of the diagram, and is made up of one sub-assembly, the hinge, and five detail parts. Each unit of assembly 9 requires 2 units of sub-assembly 4, one unit of part 1, one unit of part 2, four units of part 3, 16 units of part 6, and 24 units of part 5. The diagram shows that part 4 is an assembly, for there are other parts going into it, whereas the parts 1, 2, 3, 6, and 5 are detail parts as there are no parts entering these parts. From the diagram we can easily read the structure of sub-assembly 4.

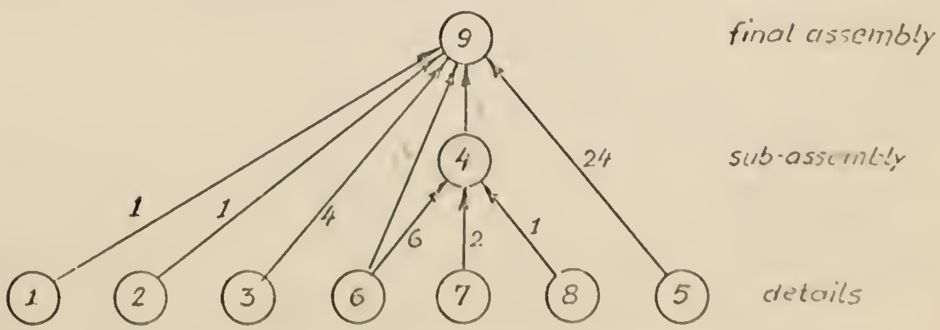


Figure 2.5. Part structure diagram of the jeweled box, example of section 2.1.

Summary. The part structure diagram illustrates the total structure of all the assemblies and final assemblies in a very clear picture. This visual representation is easy to grasp and requires few additional explanations. The part structure diagram gets too big and the overall view will be lost if we try to represent a realistic problem from industry, where hundreds of parts are required to build a final assembly. In addition, the information content of the diagram cannot be immediately used for mathematical thought and transformations.

2.33 Next assembly matrix N

The information given in the bill of material or the part structure diagram can be presented in matrix form. The bill of material of a certain assembly j can be "extended" to include every part of the production-system under consideration, even if none of these parts is needed to assemble assembly j. A quantity of zero represents every time a part listed in the bill of material, which is not required for this assembly j. The "extended part lists" may be assembled to an array, whose number of rows is equal to the total number of parts, and the number of columns is equal to the number of assemblies j. Including all parts with a zero bill of material (all entries of this bill are zero), we obtain a square matrix. The dimension of the matrix is equal to the number of parts defined by the system under consideration.

Example. The information contained in the bills of material in Figure 2.1 and the part structure diagram of Figure 2.5 is transformed into matrix form in Figure 2.6. Assembly four requires six units of item six, two units of item seven and one unit of item eight. Assembly 9 needs 1 unit of part 1, 1 unit of part 2, 4 units of part 3, 2 units of part 4, 24 units of part 5, and 16 units of part 6.

consuming parts

$$
\text{consumed parts}\quad
\begin{array}{c}
 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & [illegible] \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & [illegible] \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & [illegible] \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & [illegible] \\
0 & 0 & 0 & [illegible] & 0 & 0 & 0 & 0 & [illegible] \\
0 & 0 & 0 & [illegible] & 0 & 0 & 0 & 0 & [illegible] \\
0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & [illegible] \\
0 & 0 & 0 & [illegible] & 0 & 0 & 0 & 0 & [illegible] \\
0 & 0 & 0 & [illegible] & 0 & 0 & 0 & 0 & [illegible]
\end{pmatrix}
\end{array}
= N
$$

Figure 2.6: Next assembly matrix, N, for the jeweled box assembly. Each column represents the extended bill of material for the part indicated at the top of each column.

Definition. We define now the next assembly matrix N as a square matrix where the elements n_{ij} denote the quantity of part i directly needed to make one unit of part j. (Note: We could also define the next assembly matrix N as a rectangular matrix, not including all the zero bill of material. It is only more convenient to work with the defined "square N-matrix").

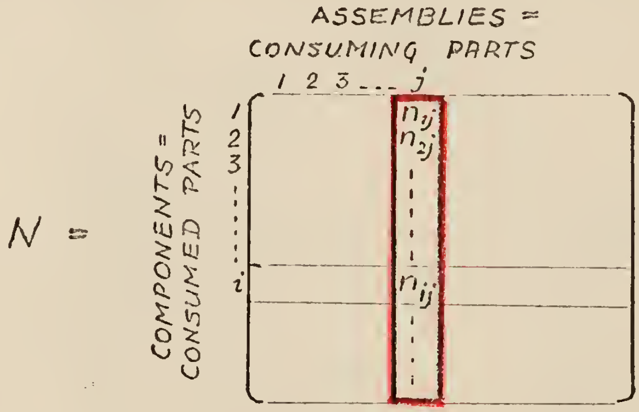


Figure 2.7. Next assembly matrix N; the red boardered column j represents the "extended bill of material" of part j.

Properties of the N-matrix.

Positive elements. All elements of the N-matrix are positive or zero. There is no meaning for negative elements n_{ij} for all n_{ij} are defined as consumed quantities.

Square matrix. Every N-matrix is square, that is the number of rows is equal to the number of columns. Every part is represented in the N-matrix as a consumed and as a consuming part. This property is a part of our definition of the N-matrix.

Triangularization. Every feasable N-matrix can be converted to a triangular matrix by interchanging of rows and columns. Proof and application of this property is shown in section 2.8.

Row. Each row i informs where and in what quantity part i is used.

Zero-row. Rows with all elements equal to zero represent the row of a final assembly or end item. These products are not assembled or used for any other part in this manufacturing system.

Column. Column j states which parts and in what quantity of these parts are directly needed to build up one part j. Each column is an "extended" bill of material for the specific part j.

Zero-column. Columns with all entries equal to zero represent detail parts or raw materials.

Main diagonal: The elements of the main diagonal n_{ij}, for all $i = j$, represents the quantity of part i which is directly consumed by itself. Generally no part is consumed by itself and all elements of the main diagonal are zero. Mathematically we call this case "non-cyclic" and write:

$$n_{ij} = 0 \quad \text{for all } i = j$$

Advantages of the matrix representation [13]. The main advantage of the matrix representation of the part structure is the conciseness and non-ambiguity compared with the graphical or any other non-mathematical formulation. The matrix algebra reduces the numerical effort to solve explosion and scheduling problems tremendously. The objective of these problems is not only to get an answer, but to obtain a result without excessive efforts. Many planning problems can be solved by enumerating and evaluating all feasable solutions. This approach - similar to the trial and error method - is rarely practical, for the number of possible solutions is astronomic in most scheduling problems.

2.34 Computer storage of the part structure

Matrix storage. The next assembly matrix N can be stored in a computer in two forms: (a) as a matrix, or (b) as an ordered list. As we will see later, the N-matrix is very "sparse", i.e. very few elements n_{ij} are non-zero most of the elements are zero. Allocating storage space to the zero elements is wasteful of computer capacity so the spanse N-matrix may be stored more efficiently as an ordered list.

Ordered list storage [27]. The next assembly matrix may be stored in two forms: ordered by columns, or ordered by rows. Both forms are kept in memory in order to facilitate rapid

retrieval by row or by column. To take advantage of the sparseness of N, only the non-zero elements are stored. Each element is stored with a row or column index, as appropriate. Moreover, a row or column begins with its own index. Thus, for the row form, we store each row N_i as

$$\{i,(j,n_{ij})\,|\,n_{ij}\neq 0,\ \forall j\} \tag{2.1*}$$

In the case that all $n_{ij} = 0$, the entire row, including the index, is omitted. Similarly, for the column form, each column is stored as

$$\{j,(i.n_{ij})\,|\,n_{ij}\neq 0,\ \forall i\} \tag{2.2}$$

The column and its index are omitted if all enties or elements are zero.

<u>Example</u>. For the N-matrix shown in Figure 2.6, the actual information stored in the computer would be for ROW ORDER:

{1:(9,1);2:(9,1);3(9,4);4:(9,2);5:(9,24);6:(4,6);(9,16);

7:(4,2);8:(4,1)}

and for COLUMN ORDER:

{4:(6,6),(7,2),(8,1);9;(1,1),(2,1),(3,4),(4,2),(5,24),

(6,16)}

The punctuation has been added for the purpose of illustration.

*The symbol $\forall$, an upside down A, stands for the term "for all".

The number standing by itself is the row index i (or the column index j), the succeeding couplets consist of column index j (or row index i) and the element n_{ij}. The matrix element n_{ij} and its associate index j (or index i) have been grouped together by parentheses.

Summary. From the above discussion, it can be seen that although matrix algebra is used to formalize the subject, the procedures are implemented with the aid of the list processing methods. Since the matrices are very sparse, the elimination of zero elements sharply reduces the amount of memory required.

2.4 TOTAL REQUIREMENT MATRIX

2.41 Introduction and Definition

In the total requirement matrix T, each element t_{ij} is defined as the total quantity of part i needed to make one unit of part j, which enters directly and/or indirectly into part j. The T-matrix can be explained in several ways. We will display here two reasonable explanations, which lead necessarily to the same result.

2.42 Explanation 1

Change the above definition slightly and state: Each element t_{ij} represents the total quantity of part i needed to assemble one unit of part j over all distances d. The definition of "distance d" was given in section 2.32.

Distance 1. The next assembly matrix, N, can be called "distance-1" quantity-matrix, for the elements n_{ij} of N are defined as the quantities of part i needed directly to build one unit of part j. According to our definition for "distance", "directly" is identically to "distance 1".

Distance 2. All parts i which are needed to produce one unit of part j at distance 2, or in other words, which are indirectly needed over one other part, can be expressed in a "distance-2" quantity-matrix. We understand, part i is at distance 2 from part j, if the part i is a direct component of a subpart k, and subpart k is a direct component of part j.

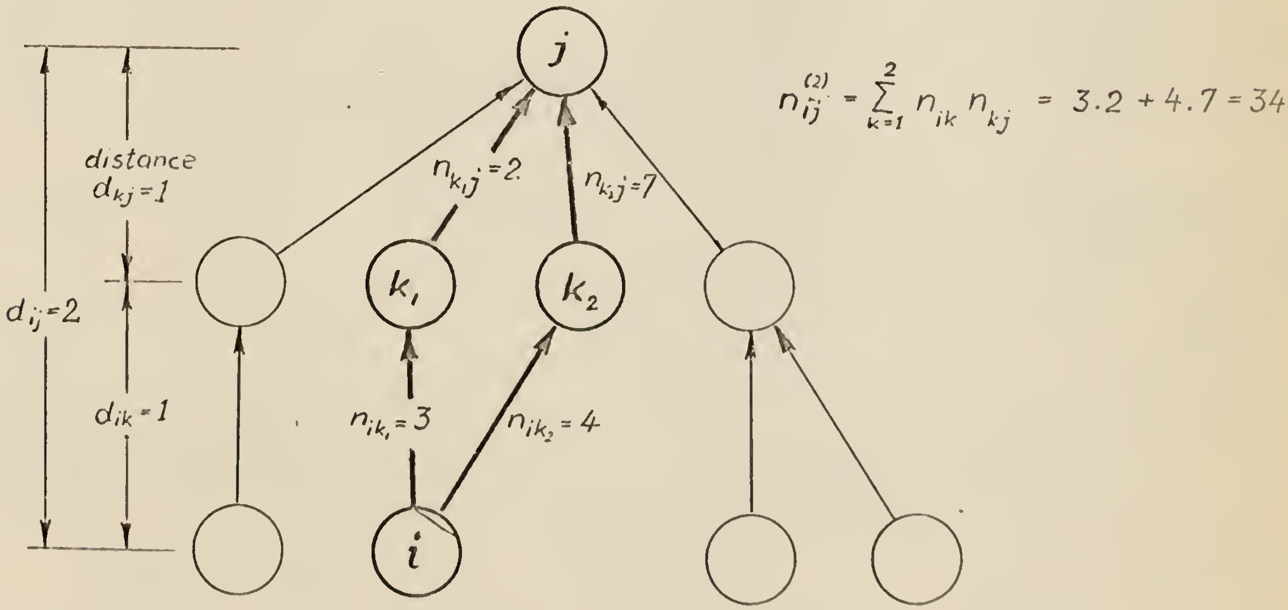


Figure 2.11. Indirect consumption at distance 2.

To calculate the total number of parts i required to build one unit of part j, we sum the products of the direct consumption of

part i into part k and the direct consumption of part k into part j. Figure 2.11 shows a brief example. For every part k_1 there are 3 units of part i required, and for every part k_2, 4 units of part i. The 2 units of part k_1 entering part j require in total 2 . 3 = 6 units of part i. The same holds true for part k_2: for all 7 units of part k_2 we need in total 4 . 7 = 28 units of part i to assemble one unit of part j. The total number of part i entering part j is therefore 3 . 2 + 4 . 7 = 34. With the illustrated problem and Figure 2.11 we can conclude: the ijth element of the "distance-2" quantity-matrix is given with

$$n_{ij}^{(2)} = \sum_k n_{ik} n_{kj} \tag{2.3}$$

changing this element into matrix form we write

$$N^{(2)} = N^2 = N.N \tag{2.4}$$

Distance k. This reasoning can be generalized for distance 3, 4, and so on. It can be proved that the "distance-k" quantity-matrix is given with N^k, where

$$N^k = N.N^{k-1} = N^{k-1}.N \tag{2.5}$$

Distance zero. Consider the requirements for a part at distance zero; the part itself is needed to give this part. To obtain one part i one part i itself is needed with zero assembly steps.

With the explanation and the previous definition we formulate:

$$n_{ij}^{(0)} = \begin{cases} 1 & \forall\ i = j \\ 0 & \forall\ i \neq j \end{cases} \tag{2.6}$$

This statement is nothing more than the definition of the identity matrix, I, consisting of "1"s in the main diagonal and zeros elsewhere.

$$N^0 = I = \begin{bmatrix} 1 & 0 & 0 & . & . & . & . & 0 \\ 0 & 1 & 0 & & & & & 0 \\ 0 & 0 & 1 & . & & & & 0 \\ . & & & & . & & & . \\ . & & & & & . & & . \\ . & & & & & & . & . \\ 0 & 0 & 0 & . & . & . & . & 1 \end{bmatrix} \tag{2.7}$$

<u>Distance $m>\delta$</u>. We denote with δ the longest distance which occurs in the part structure. All "distance-m" quantity-matrices for all $m>\delta$, are zero matrices. No part i will enter any part j at the distance $m>\delta$, for there is no distance of this magnitude in the part structure. In mathematical notation we write: in matrix format:

$$N^m = O \qquad \forall\ m>\delta \tag{2.8}$$

$$\text{where } \delta = \max \{d\}$$

or in element format:

$$n_{ij}^{(m)} = 0 \quad \forall\ i,j \text{ and } \forall\ m>\delta \tag{2.9}$$

Total requirement matrix. In the T-matrix each element t_{ij} is defined as the total quantity of part i needed to manufacture part j over all distances. Hence:

$$T = N^0 + N^1 + N^2 + \ldots + N^\delta$$

$$T = I + N + N^2 + \ldots + N^\delta \tag{2.10}$$

$$T = \sum_{k=0}^{\delta} N^k \tag{2.11}$$

This finite sum can be transformed into a closed expression, as follows:

$$T = \sum_{k=0}^{\delta} N^k \qquad = I + \sum_{k=1}^{\delta} N^k \tag{2.12}$$

$$N.T = \sum_{k=1}^{\delta+1} N^k = \sum_{k=1}^{\delta} N^k + N^{\delta+1} \qquad = \qquad \sum_{k=1}^{\delta} N^k \tag{2.13}$$

$$T - N.T = I \tag{2.14}$$

$$T(I - N) = I$$

$$T = (I - N)^{-1} \tag{2.15}$$

Numerical example. To illustrate the equation (2.11), we compute the total requirement matrix T for the "jeweled box" example. We write the N-matrix of Figure 2.6 and compute all necessary powers of the N-matrix. All powers of the N-matrix are obtained with the general matrix multiplication.

$$N^1 = \begin{array}{c|c} & \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 24 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 16 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad (2.16)$$

$$N^2 = N.N = \begin{array}{c|c} & \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad (2.17)$$

$$N^3 = N.N^2 = 0$$

$$N^4 = N.N^3 = 0$$

We recognize, that for all $m>2$, $N^m = 0$. This is not surprising for the part structure as shown in Figure 2.5, has maximal distance of $\delta = 2$. Applying equation (2.11) we write:

$$T = \sum_{k=0}^{2} N^k = I + N + N^2$$

$$T = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 24 \\ 0 & 0 & 0 & 6 & 0 & 1 & 0 & 0 & 28 \\ 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \tag{2.18}$$

Using equation (2.15) $T = (I - N)^{-1}$ we calculate first $(I - N)$ and invert it:

$$(I-N) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -4 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -24 \\ 0 & 0 & 0 & -6 & 0 & 1 & 0 & 0 & -16 \\ 0 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \tag{2.19}$$

$T = (I-N)^{-1} = T$ as above.

2.43 Explanation 2

The total consumption of part i per unit of part j is the sum of: 1) the direct consumption of part i per unit of part j and 2) the indirect consumption of all parts k, which enter

directly as subparts the part i. In mathematical terms this statement is written:

$$t^*_{ij} = n_{ij} + \sum_k t^*_{ik} \cdot n_{kj} \tag{2.20}$$

To clarify the above equation, we illustrate it with an example, showing the part structure diagram in Figure 2.12.

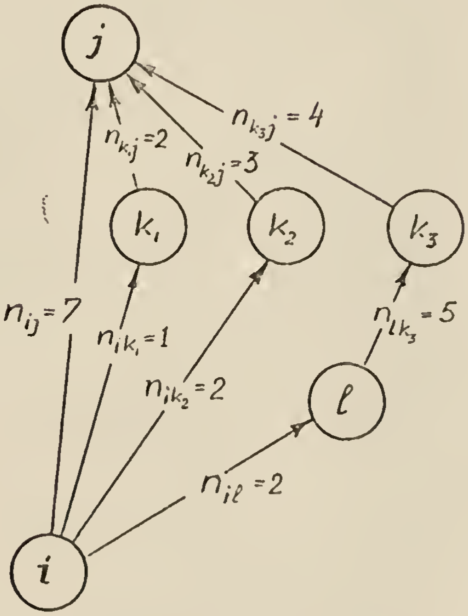


Figure 2.12. Total requirement computation 2.

Total requirements of parts i in part k:

$$t^*_{ik_1} = n_{ik_1} = 1$$

$$t^*_{ik_2} = n_{ik_2} = 2$$

$$t^*_{ik_3} = n_{il} \cdot n_{lk_3} = 2.5 = 10$$

Total requirements of parts i in part j

$$t^*_{ij} = n_{ij} + \sum_k t^*_{ik} \cdot n_{kj} =$$

$$= 7 + 1.2 + 2.3 + 10.4 = 55$$

Change from the element format in equation (2.20) to matrix format and compute the total requirement matrix T*:

$$T^* = N + T^*.N \tag{2.21}$$

$$T^* - T^*N = N$$

$$T^*.(I-N) = N$$

$$T^* = N.(I-N)^{-1} \tag{2.22}$$

In the T*-matrix defined above, we did not include the consumption of part i in part i itself, as we did in the first explanation. To obtain the same formula for the total requirements we add therefore to the computed T*-matrix the identity matrix I. According to our convention, the direct consumption or the total consumption of part i in part i is one unit, which results to be the identity matrix I:

$$n_{ij} \begin{cases} = 1 & \text{if} \quad i = j \\ = 0 & \text{if} \quad i \neq j \end{cases}$$

Thus

$$T = T^* + I \tag{2.23}$$

$$T = N.(I-N)^{-1} + I$$

$$T - I = N.(I-N)^{-1}$$

$$(T-I).(I-N) = N$$

$$T(I-N) - I = 0$$

$$T = (I-N)^{-1} \tag{2.24}$$

2.44 Discussion of the formulas for the T-matrix

From a mathematical viewpoint equation (2.15) with $T = (I - N)^{-1}$ is a closed and compact expression. However, for the actual practical computation of the T-matrix, the finite sum of $T = \sum_{k=0}^{\delta} N^k$, equation (2.11) offers less computation effort on the computer, since no matrix invertion has to be done.

2.5 CYCLIC AND NON-CYCLIC PART STRUCTURE

2.51 Introduction and definition

We introduce this section of cyclic versus non-cyclic part-structure for several reasons:

1) To define more exactly the mathematically properties of the N-matrix. The formulas and transformations, handled with this matrix definition can be applied to other mathematical models, such as business systems.

2) Clerical, key-punching or any other administrative errors lead to incorrect computations. These errors can be detected immediately as they generage a non-feasible or cyclic part structure.

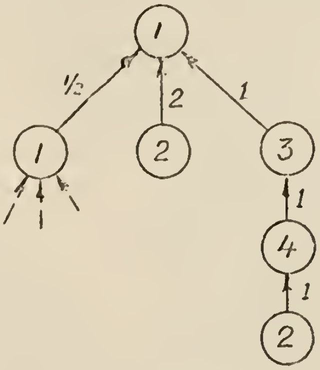
3) To enlarge our model, to set it into a larger frame, we accept not only integer quantities to be consumed, but also measurable quantities (see the definition of raw-material section (2.24)).

Cyclic bills of material or N-matrices, respectively, occur in production systems in which a produced product is needed in its own production. Parts with cyclic character cannot be manufactured, unless some quantity of each part which is consumed in itself is available to start the manufacturing process. But, once a cyclic process is started it can continue forever. Examples cover a wide range: Electricity is commonly consumed in creating the magnetic field necessary to generate the field electricity. To grow crystals (for semiconductors, etc.) a seed-crystal is required to start the process. This crystal is physically the same product. The "starter" of a dough, is necessary to begin the chemical process within the bread dough. Many agriculture processes require for the output the same product as input. A small part of the harvest is kept for use as seeds.

A part structure is said to be cyclic if there is at least one part which is consumed directly or indirectly in itself. The part structure is non-cyclic if there is no part which is consumed in itself. For the mathematically exact definition of cyclic or non-cyclic refer to the section 2.6. The part structure and the next assembly matrix N is considered "cyclic" if they were developed from a group cf bills of material with at least <u>one</u> cyclic bill of material. To determine if a N-matrix is cyclic or not and feasible or not we apply the theorems of section 2.6.

2.52 Numerical example

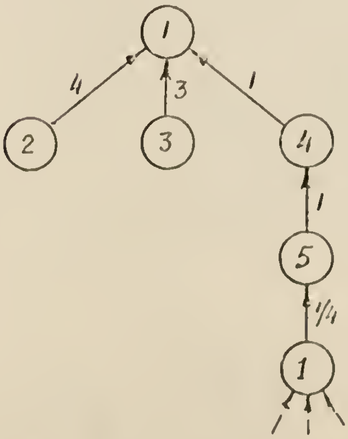
In the sample problem we assume a product 1, with the components 2, 3, and 4. The part structure diagram and the next assembly matrix N of the production system is shown in Figure 2.8. The system represented in this Figure 2.8 is said to be "direct cyclic", it requires for building part 1 that the commodity 1 is directly consumed by itself.



$$N = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Figure 2.8. Direct cyclic system.

Figure 2.9 represents an "indirect cyclic" system, for part 1 is indirectly consumed in itself.



consuming part

$$N = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{4} \\ 4 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Figure 2.9. Part structure diagram and next assembly matrix N of an indirect cyclic process.

2.53 Breaking cycles [12],[13]

In a series of problems it is possible to consider feasible cyclic processes as though they were non-cyclic. This can be done by considering a part, when it is consumed in itself, to be different from the same part when it is not consumed in itself. Continuing our example from section 2.52, we can call the electricity used to start up a generator "self-consumed electricity" or "magnetic field electricity", and the available electricity for outside load the "net generated electricity".

A cyclic next assembly matrix N can be changed into an "equivalent" non-cyclic N-matrix. A reason for making this change is to be able to triangularize with the short cut method described in section 2.7 and to apply the theory developed for the non-cyclic part structure. The simplest method to change from cyclic to non-cyclic character is to modify the designation of a part when it is consumed in itself and to assume that the "new" part has a null-bill-of-material or a zero-column, respectively. In this case each broken cycle adds a row and a column to the existing N-matrix.

For example we have a cyclic N-matrix from section 2.52:

$$N = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

This N-matrix is converted into a non-cyclic N-matrix, by treating part 1 which is consumed in part 1 as a new part 5, with a null-column or a null-bill-of-material. The new part is assigned to the fifth row and fifth column. The resulting non-cyclic N-matrix looks as follows:

$$N_{\text{non-cyclic}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

row of part "5":1/2 unit of part 5 is consumed by part "1".

null-column: no parts are required to assemble this part "5"

Figure 2.10. Converted cyclic next assembly matrix.

2.6 FEASIBILITY OF PART STRUCTURE [14], [19]

An assembly process or the next assembly matrix N, which represents this process, is feasible if all assembly steps are possible to perform.

Theorem 1. A next assembly matrix N is feasible if, and only if, the infinite sum $\sum_{k=0}^{\infty} N^k$ converges to a matrix, called the T-matrix.

Theorem 2. The N-matrix is feasible and non-cyclic if, and only if, it is nil-potent; that is, if $N^k = 0$ for some integer k less the order of N. Theorem 2 is a special case of theorem 1.

Theorem 3. The cyclic N-matrix is feasible or $T = (I - N)^{-1}$ holds true if, and only if, the infinite sum $\sum_{k=0}^{\infty} N^k$ converges. Theorem 3 is a special case of theorem 1, determining the feasibility of cyclic matrices.

2.7 UPDATING

When a manufacture performs modifications, it becomes necessary to update the N and T matrix. If such an updating would occur infrequently, it would be conceivable to compute the T-matrix from the N-matrix each time that the N-matrix is revised. In practice, however, most companies make frequent changes. A suitable shortcut for updating the T-matrix is therefore necessary for a useful application of the T-matrix in production control.

2.71 Giffler's method [13]

Let us define ΔN and ΔT as the changes of the N-matrix and of the T-matrix, respectively. Giffler proved in his paper that

$$T_1 = T + \Delta T = T.(I-\Delta N.T)^{-1} = (I-T.\Delta N)^{-1}.T \qquad (2.25)$$

Since ΔN is generally a very sparse matrix, $(I-\Delta N.T)$ and $(I-T.\Delta N)$ tend to be sparse too. The inversion of these two sparse matrices involves less time than the direct generation of the matrix $T_1 = T + \Delta T$ from $N_1 = N + \Delta N$. Even though the calculation time to determine the changed matrix is decreased with Giffler's method, the time is still very substantial.

2.72 Gleiberman's method [19]

Derivation of Formula. The formula $T = (I - N)^{-1}$ holds true for both the original matrix, N, and T, and for the changed matrix $N_1 = N + \Delta N$ and $T_1 = T + \Delta T$.

$$T = (I-N)^{-1} \tag{2.26}$$

$$T_1 = (T+\Delta T) = (I-(N+\Delta N))^{-1} = (I-N-\Delta N)^{-1} \tag{2.27}$$

Multiplying these two formulas through to eliminate the inverses gives the equations:

$$(I-N)T = I \qquad T(I-N) = I \tag{2.28}$$

$$(T+\Delta T).(I-N-\Delta N) = I \qquad (I-N-\Delta N)(T+\Delta T) = I \tag{2.29}$$

Multiplying equation (2.28) by $(T+\Delta T)$; the left equation by a left multiplication, and the right equation by a right multiplication; we get:

$$T - NT = I \qquad T - TN = I \tag{2.30}$$

$$(T+\Delta T)(T-NT) = (T+\Delta T) \qquad (T-TN)(T+\Delta T) = T+\Delta T \tag{2.31}$$

$$T^2+\Delta T.T-TNT-\Delta T.NT = T+\Delta T \qquad T^2-TNT+T\Delta T-TN\Delta T = T+\Delta T \tag{2.32}$$

Multiplying equation (2.29) by T; the left equation by a right multiplication, and the right equation by a left multiplication; we receive:

$$T+\Delta T-TN-\Delta T.N-T\Delta N-\Delta T\Delta N = I$$

$$T-NT-\Delta NT+\Delta T-N\Delta T-\Delta N.\Delta T = I \qquad (2.33)$$

$$T^2+\Delta T.T-TNT-\Delta T.NT-T\Delta NT-\Delta T\Delta NT = T$$

$$T^2-TNT-T\Delta NT+T.\Delta T-TN\Delta T-T\Delta N.\Delta T = T \qquad (2.34)$$

Subtracting equation (2.34) from equation (2.32) we have:

$$T.\Delta N.T + \Delta T.\Delta N.T = \Delta T \qquad T.\Delta N.T + T\Delta N\Delta T = \Delta T \qquad (2.35)$$

Suppose the ΔN matrix contains only non-zero elements in one row i. This means we change only the use of the item i. The structure of this part i is completely unaffected, hence, the ith column of ΔN and ΔT are zero columns.

$$\Delta T = \begin{array}{c} \left[\begin{array}{c|ccc} 0 & & & \\ 0 & & & \\ 0 & & & \\ 0 & & & \\ \vdots & & & \\ 0 & & & \end{array}\right] \\ \uparrow \text{ column i} \end{array} \qquad (2.36)$$

$$\Delta N = \left[\begin{array}{c} 0 \\ \hline \quad \\ \hline 0 \end{array}\right] \leftarrow \text{row i} \qquad (2.37)$$

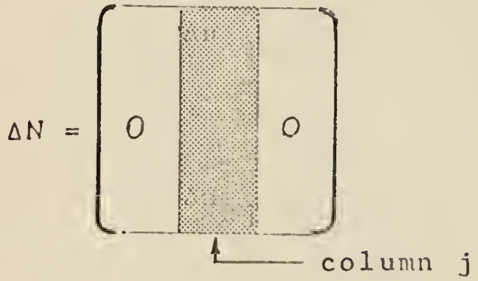
Hence,

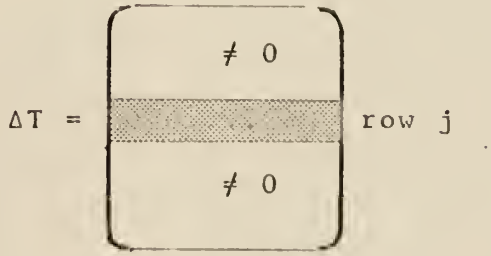
$$\Delta T.\Delta N = 0 \tag{2.38}$$

and

$$\Delta T.\Delta N.T = 0 \tag{2.39}$$

Similarly, suppose that the ΔN matrix contains only one non-zero column j. The structure of the part j is the only change for any column indicates the parts which enter the specific part j. The uses of part j are completely unaffected. Hence, the jth row of ΔT and ΔN are zero.





Hence,

$$\Delta N.\Delta T = 0 \tag{2.40}$$

$$T.\Delta N.\Delta T = 0 \tag{2.41}$$

Concluding, if ΔN has only non-zero elements in one row or in one column, the substitution of (2.39) and (2.41) into (2.35) reduces to:

$$\Delta T = T.\Delta N.T \tag{2.42}$$

The above reasoning holds if ΔN has non-zero elements in the rows of only one level or if it has non-zero elements in the columns of one level.

Application of the formula (2.42) $\Delta T = T.\Delta N.T$. Assume the case where ΔN has non-zero elements in rows and columns of different levels. For any non-zero row, i, let $\Delta_i N$ determine the matrix that is identical with ΔN in the ith row, and null elsewhere. For any non-zero column, j, let $\Delta^j N$ refer to the matrix that is identical with ΔN in the jth column, and zero elsewhere. At the first look, it seems permissable to apply equation (2.42) for each matrix.

Although this method is usually satisfactory, examples can be constructed, for which injudicious sequences of iteration steps produce troubles. In general, if any combination of elements of the original and changed matrix produces a cycle, the equation (2.42) must be applied first on the negative elements of ΔN, and then on its positive elements. This avoids the building of a cyclic matrix during the iteration process, for equation (2.26) hold only true for non-cyclic matrices (as defined in section 2.5 and 2.6).

The use of equation (2.42) is much simpler than it looks. The fact that ΔN includes only one non-zero row or column, and additionally in practice it is a very sparse row or column, makes the operation quite usable. At each iteration, it is only necessary to obtain a comparatively small number of rows and

columns. The accumulated ΔT matrix can be referred to as needed, and added to the initial T-matrix if required.

$$T_1 = T + \Delta T = T + T.\Delta N.T \tag{2.43}$$

Numerical example. The original matrix is denoted as T_o, the result of the first iteration step as T_1, of the second iteration as T_2 and so on.

$$N_0 = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \Delta N = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$T_0 = \begin{bmatrix} 1 & 0 & 3 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1 = T_0 + T_0.\Delta_2 N.T_0 = T_0 + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 3 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2 = T_1 + T_1.\Delta_5 N.T_1 = T_1 + \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 3 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

2.73 Summary

Giffler [13] suggested the generation of the T-matrix from the N-matrix by applying the updating method. Setting T = I and $\Delta N = N$, introduces an efficient method to develop the T-matrix from the N-matrix with the updating method. The question of the "cyclic" wrong way mentioned above, does not appear here, since all elements of the N-matrix are positive. The first iteration step can be performed efficiently for all rows (or columns) of one entire level. Equation (2.42) changes to:

$$\Delta T = T.\Delta_L N.T = I.\Delta_L N.I = \Delta_L N \tag{2.44}$$

$$T_1 = I + \Delta_L N \tag{2.45}$$

This method can be applied to the inversion of any matrix in the form of $(I - N)$, if N is nilpotent and has only positive and zero elements.

The methods developed in this section (2.7) are mathematically very interesting. Gleiberman's method (2.72) is a very fast and efficient way to perform updating. A periodical recalulation of the production requirements with the methods of Chapter 3 is suggested. These algorithms consider all possible modification and revisions, which occured since the last requirement calculation: appearance of new parts, change of parts in the level hierarchy, change of part usage, delayed delivery of parts etc.

2.8 N-MATRIX IN LEVEL-ORDER

An essential property of the non-cyclic N-matrix is that it can be triangularized, that is the general N-matrix can be converted into a triangular matrix (see section 2.33). The proof is: Every part may be consumed by another part, but if part i is consumed by part j, then part j cannot be consumed by part i. This is obvious, for components assembled to a certain part never require this part as a component. Mathematically this is stated with:

$$n_{ij} = 0 \quad \text{if} \quad n_{ji} > 0 \qquad \forall\, i,j \tag{2.46}$$

2.81 The triangular matrix

A triangular matrix contains only zero elements above or below the main diagonal. If the non-zero elements are above the main diagonal it is an upper triangular matrix, if the non-zero elements are below the main diagonal it is a lower triangular matrix. In mathematical terms, the triangular matrix is a lower triangular matrix if:

$$\begin{aligned} n_{ij} &= 0 \qquad \forall\, i \leq j \\ n_{ij} &\neq 0 \qquad \forall\, i > j \end{aligned} \tag{2.47}$$

and is a upper triangular matrix if:

$$\begin{aligned} n_{ij} &\neq 0 \qquad \forall\, i < j \\ n_{ij} &= 0 \qquad \forall\, i \geq j \end{aligned} \tag{2.48}$$

2.82 Triangular N-matrix

The property (see section 2.33) of the N-matrix, that every element n_{ij} is zero or positive, changes the mathematical description of the triangular matrix for our purpose:

The lower triangular N-matrix is given with:

$$
\begin{aligned}
n_{ij} &= 0 \qquad \forall\ i \leq j \\
n_{ij} &\geq 0 \qquad \forall\ i > j
\end{aligned}
\tag{2.49}
$$

or

$$
N = \begin{pmatrix} \ddots & \begin{matrix} n_{ij} = 0 \\ i \leq j \end{matrix} \\ \begin{matrix} n_{ij} \geq 0 \\ i > j \end{matrix} & \ddots \end{pmatrix}
$$

The upper triangular N-matrix is given with:

$$
\begin{aligned}
n_{ij} &\geq 0 \qquad \forall\ i < j \\
n_{ij} &= 0 \qquad \forall\ i \geq j
\end{aligned}
\tag{2.50}
$$

$$
N = \begin{pmatrix} \ddots & \begin{matrix} n_{ij} \geq 0 \\ i < j \end{matrix} \\ \begin{matrix} n_{ij} = 0 \\ i \geq j \end{matrix} & \ddots \end{pmatrix}
$$

2.83 Triangularization of the N-matrix

A feasible way of converting the general N-matrix into a triangular form is to permute rows and/or columns. Giffler [12] describes the triangularization of the N-matrix very exhaustively, including several proofs. We skip these mathematical interesting methods [12], [13], [27] and go straight to a practical explanation of the triangularization of the N-matrix.

2.84 Level order

Upon considering the different names assigned to parts, (final assembly, main-assembly, sub-assembly, assembly, and detail part) we discover a certain hierarchy. Instead of assigning an arbitrary number to the part independent of their classification we indicate the hierarchy in the part number. In other words, we sequence the parts according to their function within the part structure.

Definition. We define the level number ℓ of a part as $(d + 1)$, where d is the maximum distance of the part from all final products. The higher the level order, the lower the level number. One way to obtain a triangular N-matrix is to sort all parts according to their level. We assign each part into one of the levels, according to the maximum distance from all final assemblies. Each part will be identified with one unique part number and the level number. We use the digit before the period to

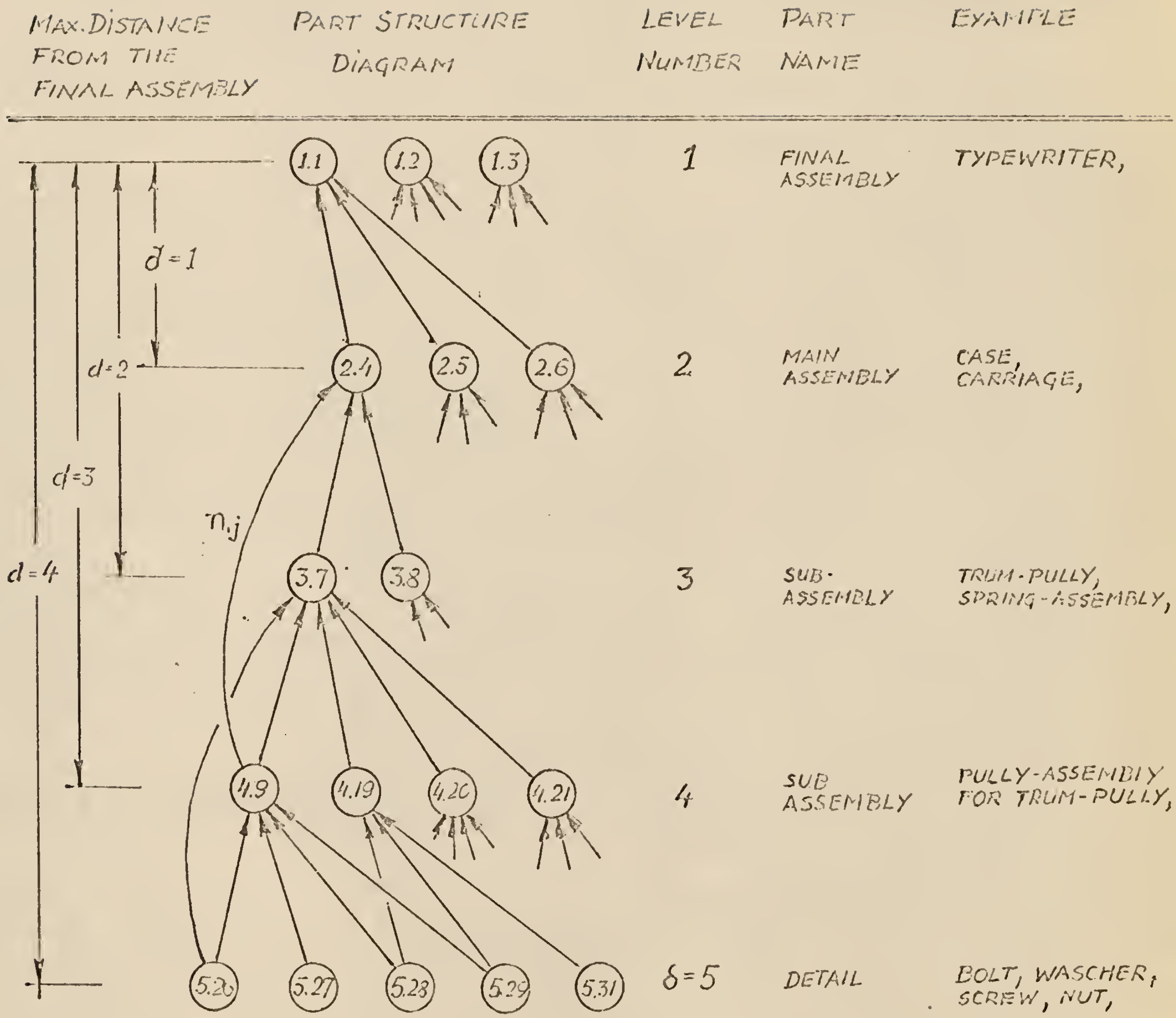


Figure 2.13. Part structure diagram and level order.

indicate the level number and the number after the period represents the unique part number in the total system. For instance the part with the part number 1088, arranged in the 3rd level will now be identified as 3.1088. We note here, that the arrangement of the part within one of the levels is inmaterial.

2.85 N-matrix in level order

According to their hierarchy, each part is assigned to a certain level. Arraying the levels in their increasing level number, we obtain an N-matrix with a lower triangular character, moreover, the matrix is block triangular. The position of any part within a certain level is unimportant.

Definition. The consumption of all parts of level ℓ per unit of parts of level k is included in the sub-matrix $N_{\ell k}$. Each element $n_{\ell.i,k.j}$ of the sub-matrix $N_{\ell k}$ denotes the quantity of part i out of the level ℓ needed directly to assemble one unit of part j out of level k.

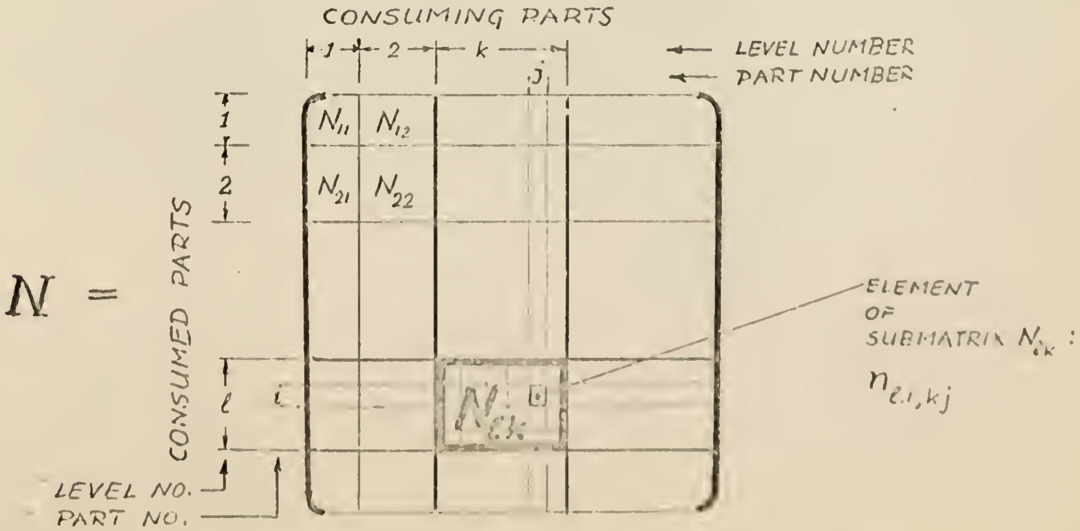


Figure 2.14. N-matrix and level order.

Lower triangular property. Every part (level ℓ) is consumed directly or indirectly by another part of higher level order, i.e. by a level with a lower level number (level k).

1. Consumption of level ℓ in k:

$$\forall\ \ell > k \qquad N_{\ell k} \begin{cases} \neq 0 & \text{if any part of level } \ell \\ & \text{is consumed by a part of level } k \\ = 0 & \text{if no single part of level } \ell \\ & \text{is consumed by level } k \end{cases} \tag{2.51}$$

2. No consumption of level ℓ in k:

$$\forall\ \ell < k \qquad N_{\ell k} = 0 \tag{2.52}$$

3. No consumption of two parts from the same level:

$$\forall\ \ell = k \qquad N_{\ell k} = 0 \tag{2.53}$$

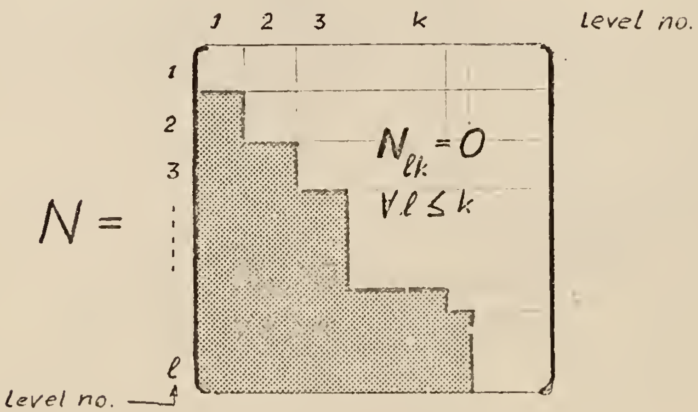


Figure 2.15. Picture of N-matrix in level order.

Block triangular property. The block character is obtained by grouping all parts with the same assembly function into the same

level. All parts of a certain level are consumed by parts of another level (with a lower level number); none of the parts are consumed by parts of the same level.

Advantage of level order.

** The main advantage of the N-matrix in level order is its triangular character. Combined with this property, all the advantages of the triangular matrix are gained.

** The accumulation of all information carrying elements under the main diagonal requires less storage space. For a triangular N-matrix of the order n, there are only $(n^2/2 - n/2) = (n - 1).\,n/2$ elements necessary to describe the total situation instead of n^2 elements with the general N-matrix.

** Matrix invertion and several other matrix operations may be performed by a shorter and faster method. This is illustrated in the computation of the total requirement matrix T (see section 2.86) as well as of the requirement matrix G (see section 3.3).

** The introduction of the level order is based on the advantage of the calculation of the requirement matrix G without the cumbersome and time consuming computation of the total requirement matrix T. See section 3.3.

** The application of level order reduces clerical errors, caused by interchanging of consuming with consumed parts and vice versa.

** The level order offers an excellent survey of the position of a part within the part hierarchy. The suggested part number insures a fast and secure method of finding and locating any part.

2.86 Computation of the T-matrix with a triangular N-matrix [13]

The basic equation for determining the elements of $(I - N)^{-1}$ when N is triangular and non-negative are:

for lower triangularity (as we defined the N-matrix in level order):

$$t_{ij} \begin{cases} = \delta_{ij} = \sum_{k=j}^{i-1} n_{ik} t_{kj} & \forall\ j \leq i \\ = 0 & \forall\ j > i \end{cases} \tag{2.54}$$

or

$$t_{ij} \begin{cases} = \delta_{ij} + \sum_{k=i+1}^{m} t_{ik} n_{kj} & \forall\ j \leq i \\ = 0 & \forall\ j > i \end{cases} \tag{2.55}$$

for upper triangularity:

$$t_{ij} \begin{cases} = \delta_{ij} + \sum_{k=j}^{i-1} t_{ik} n_{kj} & \forall\ j \geq i \\ = 0 & \forall\ j < i \end{cases}$$

or

$$t_{ij} \begin{cases} = \delta_{ij} + \sum_{k=i+1}^{m} n_{ik} t_{kj} & \forall\ j \geq i \\ = 0 & \forall\ j < i \end{cases}$$

where δ_{ij} is the Kronecker-symbol, defined with $\delta_{ij} = 1$ for all $i = j$ and $\delta_{ij} = 0$ for all $i \neq j$.

Derivation of formula (2.54). The basic formula to calculate the T-matrix is given with (2.15):

$$T = (I - N)^{-1}$$

$$(I - N).T = I \qquad\qquad T(I - N) = I$$

$$T - N.T = I \qquad\qquad T - TN = I$$

$$T = I + N.T \qquad\qquad T = I + TN \tag{2.56}$$

or in element format:

$$t_{ij} = \delta_{ij} + \sum_{k=1}^{m} n_{ik}t_{kj} \qquad\qquad t_{ij} = \delta_{ij} + \sum_{k=1}^{m} t_{ik}n_{kj} \tag{2.57}$$

From the lower triangular property of the N-matrix, equation (2.49) and the definition of T we know:

$$n_{ij} = 0 \ ; \ t_{ij} = 0 \qquad \forall \ i \leq j \tag{2.58}$$

that allows one to change (2.57) to:

$$t_{ij} \begin{cases} = \delta_{ij} + \sum_{k=j}^{i-1} n_{ik}t_{kj} & \forall \ j \leq i \\ = 0 & \forall \ j > i \end{cases}$$

which is equation (2.54). Similarly, for (2.55), for N.T = T.N from line (2.56).

Remarks. Using equation (2.54) the T-matrix is economically constructed from the columns, column by column from left to right. If the columns of t_{ij} are determined in this suggested

sequence, it will be seen that each newly calculated t_{ij} may be stored directly over n_{ij} (in the computer), for n_{ij} will not be required anymore to calculate the remaining t_{ij}. On the other hand, each calculated t_{ij} is required in the next iteration step in the next row.

Example. For illustration purpose we use equation (2.54) to calculate T from a lower triangular N-matrix.

$$N = \begin{pmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 4 & 0 \end{pmatrix}$$

column 1: $j = 1$

$$t_{11} = \delta_{ij} \qquad\qquad = 1$$

$$t_{21} = 0 + \sum_{k=1}^{1} n_{2k} t_{k1} = 2.1 \qquad = 2$$

$$t_{31} = 0 + \sum_{k=1}^{2} n_{3k} t_{k1} = 3.1 + 4.2 = 11$$

column 2: $j = 2$

$$t_{12} = 0 \qquad\qquad = 0$$

$$t_{22} = \delta_{ij} \qquad\qquad = 1$$

$$t_{32} = 0 + \sum_{k=2}^{2} n_{3k} t_{k2} = 0+4.1 \qquad = 4$$

column 3: $j = 3$

$$t_{13} = 0 \qquad\qquad = 0$$

$$t_{23} = 0 \qquad\qquad = 0$$

$$t_{33} = \delta_{ij} \qquad\qquad = 1$$

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 11 & 4 & 1 \end{pmatrix}$$

3. DEMAND PROBLEM

A sales forecast or a customer order generates a certain demand for final assemblies. This demand in turn causes a demand for all the main-assemblies, which enter these final assemblies, and each demand for a main-assembly generates a demand for all the entering parts. It is necessary to determine the quantity of each part that must be manufactured (for assemblies) or purchased (for the detail parts) to meet a given demand and call it "requirement". This demand may exist for any part, for in order to satisfy customers, a manufacturer generally sells final products as well as spare parts.

We define "DEMAND" as the quantity of a part needed at a certain time, t_1 for the market. "REQUIREMENT" will be used for the quantity of a part which has to be produced (for assemblies) or ordered (for detail parts) at time t_0 to meet the given "demand" at time t_1.

3.1 TYPES OF DEMAND

We discuss first a sequence of simple demand problems. Step by step we will enlarge the simple problem to solve the realistic time-dependent demand problem spanning several future time periods. This chapter discusses the following demand problems:

1. simple demand,
2. simple demand and level order,
3. netted demand, edited demand, or simple demand with available inventory,

4. reorder demand,

5. time-dependent demand, and

6. time-dependent demand with available inventory.

3.2 SIMPLE DEMAND

Simple demand requires only the computation of the quantity of every part to fulfill the desired parts for the market. There is no indication of when the part production has to be started or finished, and no inventory is available.

Definition. We define the column vector D as the demand vector, where the ith element d_i is the demand quantity of part i. Since t_{ij} units of part i are necessary to build one unit of part j, the total quantity to be produced or purchased of part i to fulfill the demand d_j of part j is given by the product $t_{ij} . d_j$. The total requirements g_i of part i, satisfying all demands of all parts j = 1,..., n, is $\sum_j t_{ij}.d_j$. Hence, the basic equation for determining the requirements is

$$g_i = \sum_j t_{ij}.d_j \qquad (3.1)$$

or in matrix format

$$G = T.D = (I - N)^{-1}.D \qquad (3.2)$$

Definition. Column vector G is the requirement vector, where the ith element g_i represents the total number of units of part i that must be manufactured (for assemblies) or purchased (for details) to meet the demand set by the demand vector D.

Discussion of the formula G = T . D. To calculate the requirement vector G with equation (3.2) it is necessary to do time consuming inversion of (I - N). Different methods [10] have been developed to avoid the computation of the T-matrix and to find a faster method.

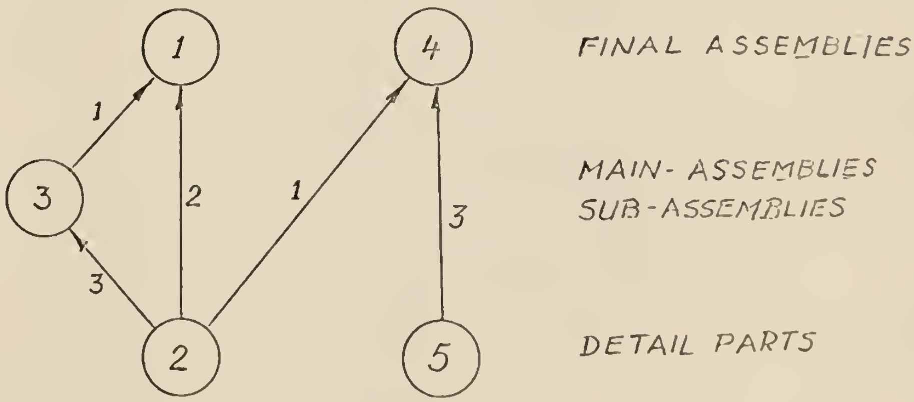


Figure 3.1. Part structure diagram, all parts are arranged in level order, but numbered arbitrary.

$$
\begin{array}{ccc}
 & & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \leftarrow \text{Part No.} \\
N = & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 3 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \end{bmatrix}
\end{array}
$$

To Figure 3.2.

$$T = \sum_{k=1}^{2} N^k = I + N^1 + N^2 = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \leftarrow \text{Part No.} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 5 & 1 & 3 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 & 1 \end{pmatrix} \end{array}$$

Figure 3.2. N-matrix and T-matrix of the part structure in Figure 3.1. The parts are arranged in the sequence of the part numbers and are not in level order.

Example. Let us assume, we know the demand for the parts shown in Figure 3.1 and 3.2. Final assemblies, as well as other assemblies and detail parts required as spare parts, should be ready on a certain day. We are asked to deliver 100 units of final assembly 1 and 30 units of final assembly 4 and 10 units of sub-assembly 3. Thus:

$$D = \begin{pmatrix} 100 \\ 0 \\ 10 \\ 30 \\ 0 \end{pmatrix}$$

In the example the matrix multiplication G = T . D gives:

$$G = T.D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 5 & 1 & 3 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 & 1 \end{pmatrix} . \begin{pmatrix} 100 \\ 0 \\ 10 \\ 30 \\ 0 \end{pmatrix} = \begin{pmatrix} 100 \\ 560 \\ 110 \\ 30 \\ 90 \end{pmatrix}$$

The vector G specifies the unit quantity of each part required to meet the order D.

3.3 SIMPLE DEMAND AND LEVEL ORDER.

To compute the simple requirement vector G for the demand D with the equation $G = T \cdot D$, it is essential to know the total requirement matrix T. The formula $G = T \cdot D$ is short and looks very elegant, however, the computation of the T-matrix embraces the inversion of $(I - N)$. An average end item, such as a typewriter or an engine holds thousands of components, and the N or T-matrix has as many columns and rows as there are components in the total production system. The inversion of a matrix of this size consumes tremendous time, even on large scale computers. We want to avoid the computation and storage of the T-matrix in the computer. We compute the requirements level by level starting with level 1 with the following equations:

$$\text{Level 1:} \quad G_1 = D_1 \tag{3.3}$$

$$\text{Level 2:} \quad G_2 = D_2 + N_{21} \cdot G_1 \tag{3.4}$$

$$\text{Level 3:} \quad G_3 = D_3 + N_{31}\, G_1 + N_{32} \cdot G_2 \tag{3.5}$$

$$\text{Level } \ell\text{:} \quad G_\ell = D_\ell + \sum_{k=1}^{\ell-1} N_{\ell k} \cdot G_k \tag{3.6}$$

$$\ell = 1, \ldots, \delta$$

$$\text{where } \delta = \max\{\ell \mid N^{\ell+1} = 0\}$$

Subvector. The indexed vectors represent subvectors of the G or D-vector, respectively. The subscript indicates the level of the specific subvector, which shows the demand or the requirement of this level. Vector D_ℓ, for instance, stands for the demand of all parts of level ℓ.

$$D = \left[\begin{array}{c} d_{1.1}\\ d_{1.2}\\ \vdots\\ d_{1.u}\\ \hline d_{2.10}\\ d_{2.11}\\ \vdots\\ d_{2.v}\\ \hline \vdots\\ \hline d_{\delta.12}\\ d_{\delta.13}\\ \vdots\\ d_{\delta.w}\end{array}\right] = \left[\begin{array}{c} D_1 \\ \hline D_2 \\ \hline \vdots \\ \hline D_\delta \end{array}\right] \begin{array}{l} \left.\right\} \text{subvector } D_1, \text{ demand of "level-1" parts} \\[2em] \left.\right\} \text{subvector } D_2, \text{ demand of all "level-2" parts} \\[2em] \left.\right\} \text{subvector } D_\delta \text{ demand of all "level-}\delta\text{" parts (detail parts)} \end{array} \tag{3.7}$$

The subvectors may be written in two fashions:

1. As shown above, where the subvector contains only the elements of level ℓ:

$$D_\ell \quad \begin{bmatrix} d_{\ell.1} \\ d_{\ell.2} \\ \vdots \\ d_{\ell.m} \end{bmatrix} \tag{3.8}$$

2. Where D_ℓ has as many elements as D, but all elements other than level ℓ are zero.

$$D_\ell = \left[\begin{array}{c} \left.\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \end{array}\right\} \text{all elements for level } 1, \ldots, \ell-1 \text{ are zero.} \\ \hline \left.\begin{array}{c} d_{\ell.1} \\ d_{\ell.2} \\ \vdots \\ d_{\ell.} \end{array}\right\} \text{demand of all "level-}\ell\text{" parts} \\ \hline \left.\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \end{array}\right\} \text{all elements for level } \ell+1, \ldots, \delta \text{ are zero} \end{array}\right]$$

<u>Explanation of equation (3.6)</u>. To satisfy the demand D_1 of level 1 we produce $G_1 = D_1$ parts of level 1. This demand D_1 in turn generates a demand of parts of a lower level, which enter into the parts of level 1. The required parts of level 1 is simply

$$G_1 = D_1$$

The required parts of level 2 is the sum of the demand D_2 for level 2 plus all the parts of level 2 needed in level 1. For every part j in level 1 $n_{2.i,1.j}$ parts i of level 2 are required. Hence, $N_{21.} G_1$ parts of level 2 are necessary to satisfy the requirements G_1 of level 1.

$$G_2 = D_2 + N_{21} G_1$$

The required parts of level 3 are the sum of the direct demand D_3 of "level-3" parts plus all the level-3 parts needed to satisfy the requirement of level 1 and 2. For every part j of level 3, n_{ij} parts i are needed of level 1 or 2, respectively. Hence, $N_{31} \cdot G_1$ parts of level 3 are required to be produced to satisfy the requirement of demand G_1 of level 1, and $N_{32}.G_2$ parts of level 3 are necessary to be manufactured to satisfy the requirement G_2 of level 2.

$$G_3 = D_3 + N_{31} G_1 + N_{32} G_2$$

For level ℓ, in general, we are asked to produce the sum of the direct demand D_ℓ plus all "level-ℓ" parts which are required in level one up to and including$(\ell-1)$.

$$G_\ell = D_\ell + \sum_{k=1}^{\ell-1} N_{\ell k}.G_k$$

The requirement vector G is assembled from the δ sub-vectors G_ℓ, as equation (3.8).

$$G = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_\delta \end{bmatrix} \tag{3.10}$$

Proof of equation (3.6). We show with the following proof that the equation (3.6) for the requirement computation is identical to formula (3.2) G = T.D. The N-matrix in level order is lower triangular:

$$N_{\ell k} = 0 \qquad \forall\ \ell \leq k \tag{3.11}$$

$$N_{\ell k} \geq 0 \qquad \forall\ \ell > k$$

Hence, the equations (3.6) can be rewritten with

$$G_\ell = D_\ell + \sum_{k=1}^{\ell-1} N_{\ell k}\ G_k + \sum_{k=\ell}^{\delta} N_{\ell k}\ G_k = D_\ell + \sum_{k=1}^{\delta} N_{\ell k}\ G_k \tag{3.13}$$

The values of the G_ℓ-vectors did not change, for the added summands include the factors $N_{\ell k}$ ($\ell \leq k$) which are all zero. We continue:

$$G = D + N.G \tag{3.14}$$

$$G - N.G = D$$

$$G = (I-N)^{-1}.D = T.D \tag{3.15}$$

The result, equation (3.15) is identical to equation (3.2) of section 3.2.

Example. To illustrate the equations we calculate the requirements for a given demand with a part structure in three levels, see Figure 3.3.

Demand, given with the demand vector D:

$$D = \left[\begin{array}{c} D_1 \\ \hline D_2 \\ \hline D_3 \end{array}\right] = \left[\begin{array}{c} 100 \\ 150 \\ \hline 0 \\ 0 \\ 22 \\ 12 \\ \hline 10 \\ 20 \end{array}\right] \begin{array}{l} \left.\begin{array}{c} \\ \\ \end{array}\right\} \text{1. Level} \\ \left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} \text{2. Level} \\ \left.\begin{array}{c} \\ \\ \end{array}\right\} \text{3. Level} \end{array}$$

Part structure, given with the next assembly matrix N:

$$N = \left[\begin{array}{c|c|c} N_{11} & N_{12} & N_{13} \\ \hline N_{21} & N_{22} & N_{23} \\ \hline N_{21} & N_{32} & N_{33} \end{array}\right] = \left[\begin{array}{cc|cccc|cc} & 0 & & 0 & & & 0 & \\ \hline 3 & 0 & & & & & & \\ 1 & 4 & & 0 & & & 0 & \\ 2 & 7 & & & & & & \\ 2 & 1 & & & & & & \\ \hline 5 & 7 & 5 & 3 & 0 & 0 & & 0 \\ 8 & 9 & 9 & 0 & 0 & 3 & & \end{array}\right]$$

Figure 3.3. Next assembly matrix N.

Requirements, computation of the requirement vectors G_ℓ with equation (4.6):

$$G_1 = D_1 = \begin{bmatrix} 100 \\ 150 \end{bmatrix} = \begin{bmatrix} 100 \\ 150 \end{bmatrix}$$

$$G_2 = D_2 + N_{21}G_1 = \begin{bmatrix} 0 \\ 0 \\ 22 \\ 12 \end{bmatrix} + \begin{bmatrix} 3 & 0 \\ 1 & 4 \\ 2 & 7 \\ 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 100 \\ 150 \end{bmatrix} = \begin{bmatrix} 300 \\ 700 \\ 1072 \\ 362 \end{bmatrix}$$

$$G_3 = D_3 + N_{31}G_1 + N_{32}G_2 = \begin{bmatrix} 10 \\ 20 \end{bmatrix} + \begin{bmatrix} 5 & 7 \\ 8 & 9 \end{bmatrix} \cdot \begin{bmatrix} 100 \\ 150 \end{bmatrix} + \begin{bmatrix} 5 & 3 & 0 & 0 \\ 9 & 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 300 \\ 700 \\ 1072 \\ 362 \end{bmatrix} = \begin{bmatrix} 5160 \\ 5956 \end{bmatrix}$$

Requirements, in form of the requirement vector G, assembled from the sub-vectors G_ℓ; $\ell = 1, 2, 3$;

$$G = \begin{bmatrix} G_1 \\ \hline G_2 \\ \hline G_3 \end{bmatrix} = \begin{bmatrix} 100 \\ 150 \\ \hline 300 \\ 700 \\ 1072 \\ 362 \\ \hline 5160 \\ 5956 \end{bmatrix}$$

3.4 NETTED OR EDITED DEMAND

The "netted demand" problem is an enlargement of the simple demand problem stated in section 3.3. We assume the existance of an inventory of the items demanded. The original gross requirements must be adjusted against the available inventory, i.e., "netted." Whenever the requirement of an assembly is adjusted against the inventory (in other words, reduced by the available stock) the requirement for parts of a lower level will change. The objective of "netted requirements" is to take as many items as possible from the existing inventory.

3.41 Giffler's approach

Giffler [13] formalized the netted demand problem with a linear programming program. He calculated first the gross requirements (or as we called it in section 3.3 the simple requirements, G) for all parts, independent of their level status; and then reduced gross requirements via the linear programming method by the available inventory.

The object function to be maximized is formulated as:

$$F(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_j \, t_{ij} \tag{3.16}$$

where x_j is the amount of part j taken from the inventory. F(X) is the total quantity of all parts which are taken from the

available inventory. The factor t_{ij} stands for the ith element of the total requirement matrix T. The <u>constraints</u> for all the x_j's are given by two sets of matrix equations:

$$X \leq A \qquad (3.17)$$

$$T.X \leq G \quad \text{with} \quad G = T.D \qquad (3.18)$$

where X is the column vector, representing the inventory supply, with x_j as the jth element, the quantity of part j taken from the inventory. The first constraint (3.17) insures that the inventory supply x_j never exceeds the available inventory a_j. We cannot reduce the inventory by more than what is available. The second constraint (3.18) insures that the total quantity of part j which was produced in the manufacturing of X will not exceed G = T . D.

3.42 Loewner's approach [27]

The technique in section 3.42 requires the computation of the matrix T and the G-vector, connected by a linear program. To avoid the substantial calculations of these matrices and the linear program, we work with the N-matrix in level order. To supply the demand with as many items as possible from the existing inventory, the netting operation will be started with the final assemblies, continuing with the next lower level, and so on, going down level by level. The following sections basically use Loewner's ideas [27].

3.43 Definitions

For this and the following section, the computation of the netted requirements, we use the following notations.

The available finished parts, the inventory, are represented by the availability vector, A, where the element a_i shows the number of parts i available from stock. The original available stock will be denoted with $A^{(0)}$, and the remaining available stock after reduction or the netting operation will be called $A^{(1)}$. Note, the superscripts (0) or (1) indicate the stage of the inventory vector, and will always be enclosed by parentheses. The exponents are written without parentheses, such as $(N)^2$ or N^2.

The requirements before the netting operation are called gross requirements and are denoted with the column vector G. If no inventory would be available, these quantities g_i have to be produced.

The requirement, that is to be manufactured, is represented with the edited or netted requirement vector R. The element r_i of this column vector gives the number of parts i that have to be fabricated to cover the demand D.

In all vectors the parts are arranged in level order. The whole column vector is split into δ subvectors V_ℓ, where V_ℓ defines the demand, inventory, and requirements, respectively, of level ℓ. The total number of levels is given with δ, and m_ℓ denotes the number of parts in level ℓ. Note, that $\sum_{k=1}^{\delta} m_k = n$, is the total number of parts in the system.

3.44 Mathematical formulation

We use the same basic idea and equations of section 3.3. The requirements are computed in the same way, level by level, as the sum of the direct demand plus the indirect demand required in parts of higher level (lower level number). Every time we evaluated the gross requirements for one level ℓ, we reduce this quantity by the available inventory A_ℓ. Then we continue with the next lower level ($\ell+1$). This method yields the correct quantity of parts to be produced. In the lower level ($\ell+1$), only the quantities of the sub-assemblies required for the netted requirements of level ℓ will be considered and eventually manufactured. We state:

Level 1: $$R_1 = \{D_1\} \ominus A_1 \tag{3.19}$$

Level 2: $$R_2 = \{D_2 + N_{21} R_1\} \ominus A_2 \tag{3.20}$$

Level 3: $$R_3 = \{D_3 + N_{31} R_1 + N_{32} R_2\} \ominus A_3 \tag{3.21}$$

$\vdots$

Level ℓ: $$R_\ell = \underbrace{\underbrace{\{\underbrace{D_\ell}_{\substack{\text{direct} \\ \text{demand } D_\ell}} + \underbrace{N_{\ell 1} R_1 + N_{\ell 2} R_2 + \dots + N_{\ell,\ell-1} R_{\ell-1}\}}_{\substack{\text{indirect} \\ \text{demand } I_\ell}}}_{\text{gross requirement } G_\ell} \ominus \underbrace{A_\ell}_{\substack{\text{available} \\ \text{inventory} \\ A_\ell}}}_{\text{net requirement } R_\ell}$$

In general we write:

$$R_\ell = \{D_\ell + \sum_{k=1}^{\ell-1} N_{\ell k} R_k\} \ominus A_\ell \tag{3.22}$$

$$\forall\ \ell = 1, \ldots, \delta$$

$$\text{where } \delta = \max\{\ell \mid N^{\ell+1} = 0\}$$

The diminish symbol $\ominus$ is defined as:

$$a \ominus b = \begin{cases} a - b & \text{if } a > b \\ 0 & \text{if } a \leq b \end{cases} \tag{3.23}$$

Block diagram. The practical application and computation of equation (3.22) is schematically illustrated with the block diagram in Figure 3.6. For the computer calculation of the netted requirement we choose technique two, see equation (3.9). Every subvector V_ℓ has as many elements as the vector V, only that all elements other than from the indicated level ℓ are zero. A similar notation is introduced for the N-matrix. The submatrix N_ℓ of the N-matrix has the same number of rows and columns as the N-matrix, only all rows other than that of level ℓ are zero rows. Figure 3.4 and 3.5 illustrate the explanation. The requirement vector R expands with each step by one level as it is computed. At the start of step ℓ, vector R contains only elements in level-1 upto and including $(\ell-1)$, and only zero-elements in the levels ℓ to δ. At the end of step ℓ the vector is increased by the new elements in the level ℓ.

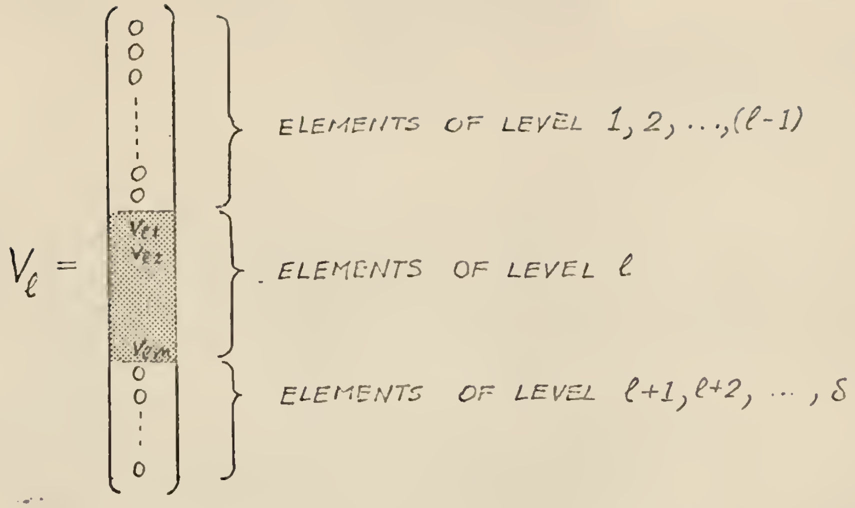


Figure 3.4. Subvector V_ℓ of vector V.

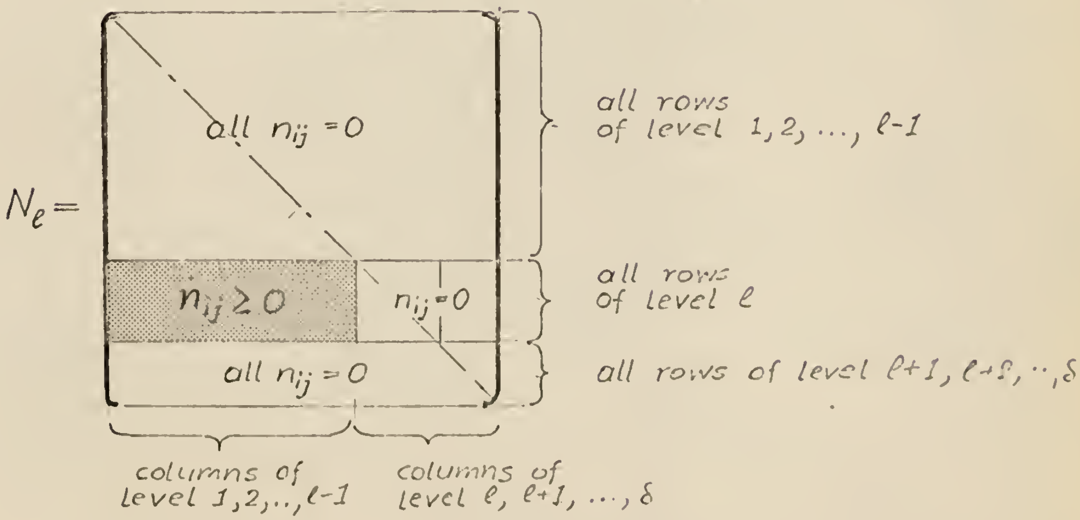


Figure 3.5. Submatrix N_ℓ of N-matrix.

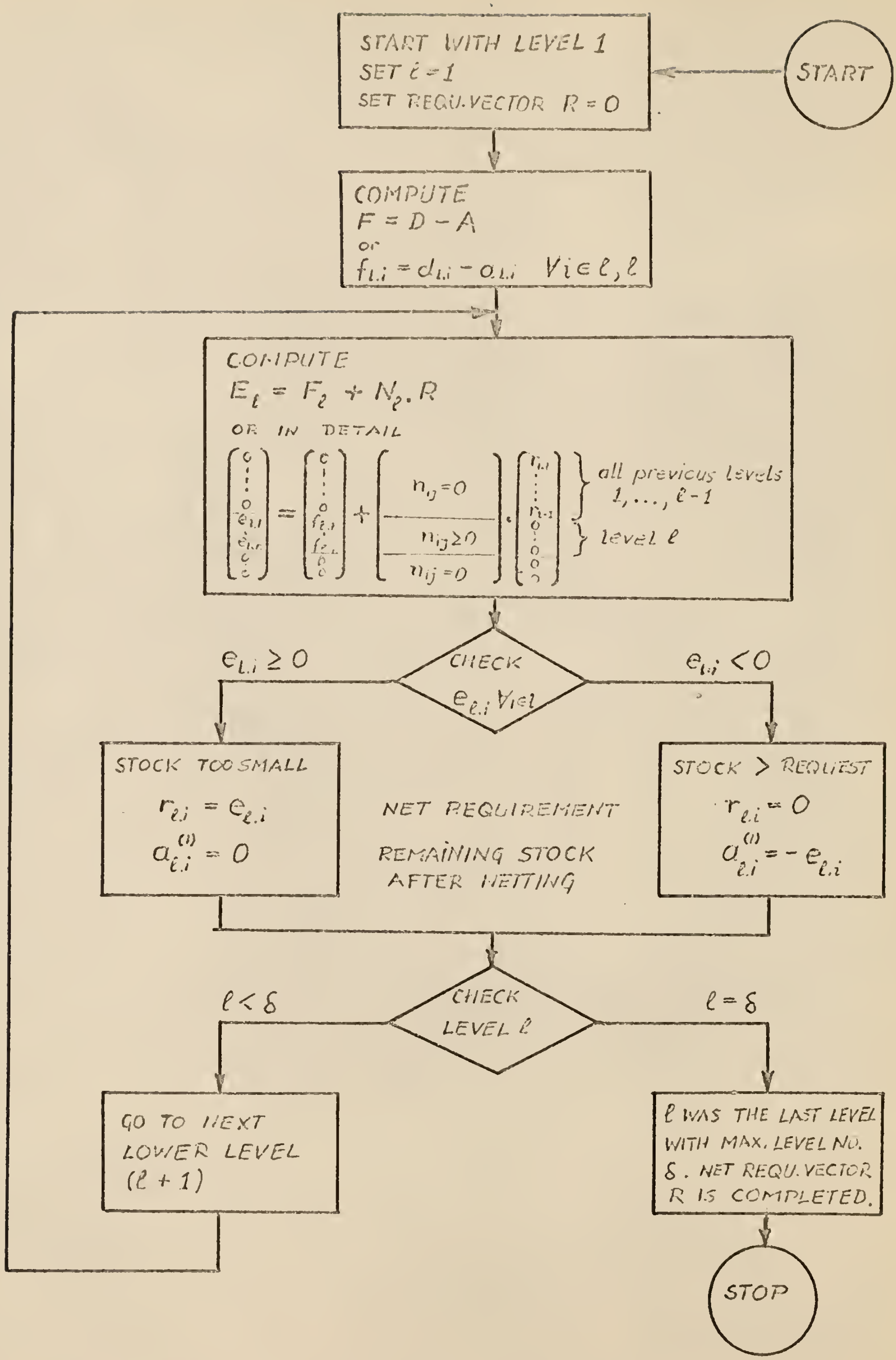


Figure 3.6. Blockdiagram for netted requirements.

3.5 REORDER DEMAND [27]

This section presents a refinement of the simple requirements with available inventory, shown in section 3.4. The inventory is designed to keep stock available, whenever possible, and to give orders to production (for assemblies) and to purchasing (for detail parts and raw material) when the inventory stock falls under a certain level. The order-releasing stock level is called a "reorder point". All demands or requirements from the salesforecast are covered from inventory if possible. The production replentishes the inventory whenever necessary with the following procedure.

Review [35]. This refined reorder system, usually called the "Q-system", is designed for the dynamic inventory problem under risk with two features: 1) there is a number of orders possible for the future and 2) there is a known probability distribution of future demand. This control system has a fixed order size q and a varying order period p. Whenever the stock in the inventory falls to a certain minimal level, an order is automatically placed for the predetermined fixed amount q. The specified minimum level - the reorder point k - is based on the time lag between order and delivery of the item. The calculations for the reorder point k and the reorder quantity q are given by M. Starr and D. Miller [35].

Time in the general model is continuous. However, our model working with planning periods of a certain length, requires discrete units for time.

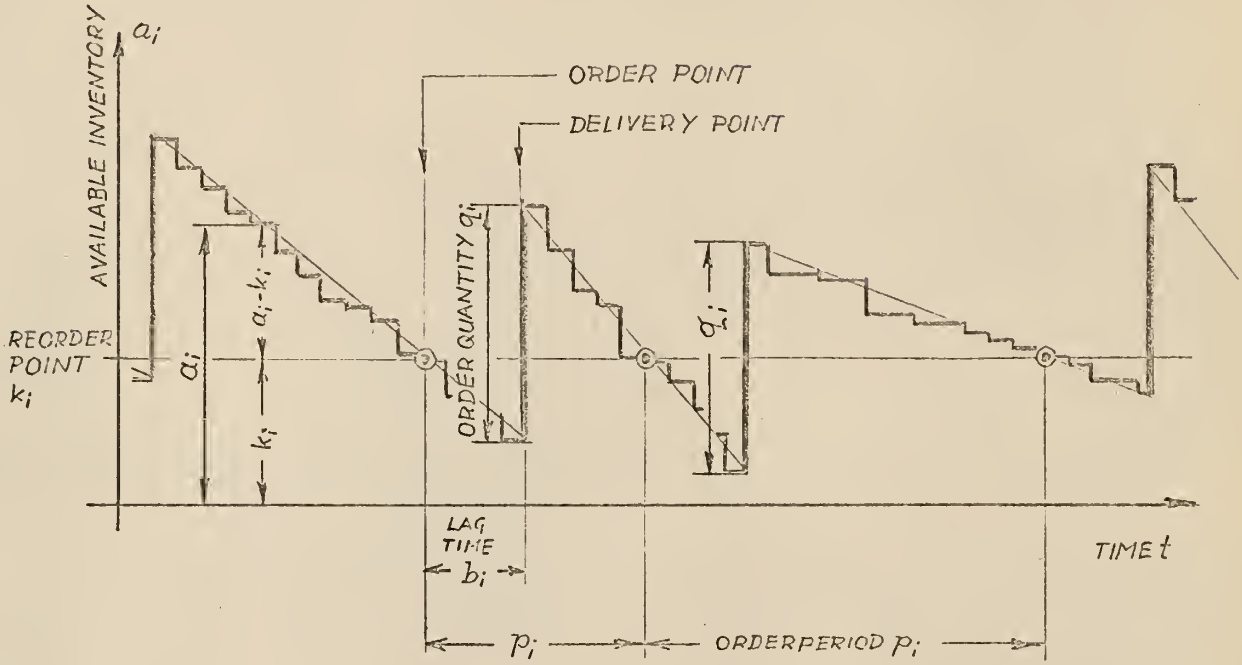


Figure 3.7. Relation of inventory usage, reorder and time for the Q-system. The index i stands for part i. In this Figure the demand rate is assumed to be constant for one order period. This is done for the sake of an easier understanding. The demand rate can fluctuate over the time.

Definition. We denote as K the vector of the reorder point, with the (i)th element k_i as the reorder point of part i. Q is defined as the vector of the order quantity, where the (i)th element q_i represents the order quantity of part i.

Algorithm. The previous algorithm of the netted requirement in section 3.4 is changed slightly. Without a detailed explanation we show the modified block diagram in Figure 3.8 and refer

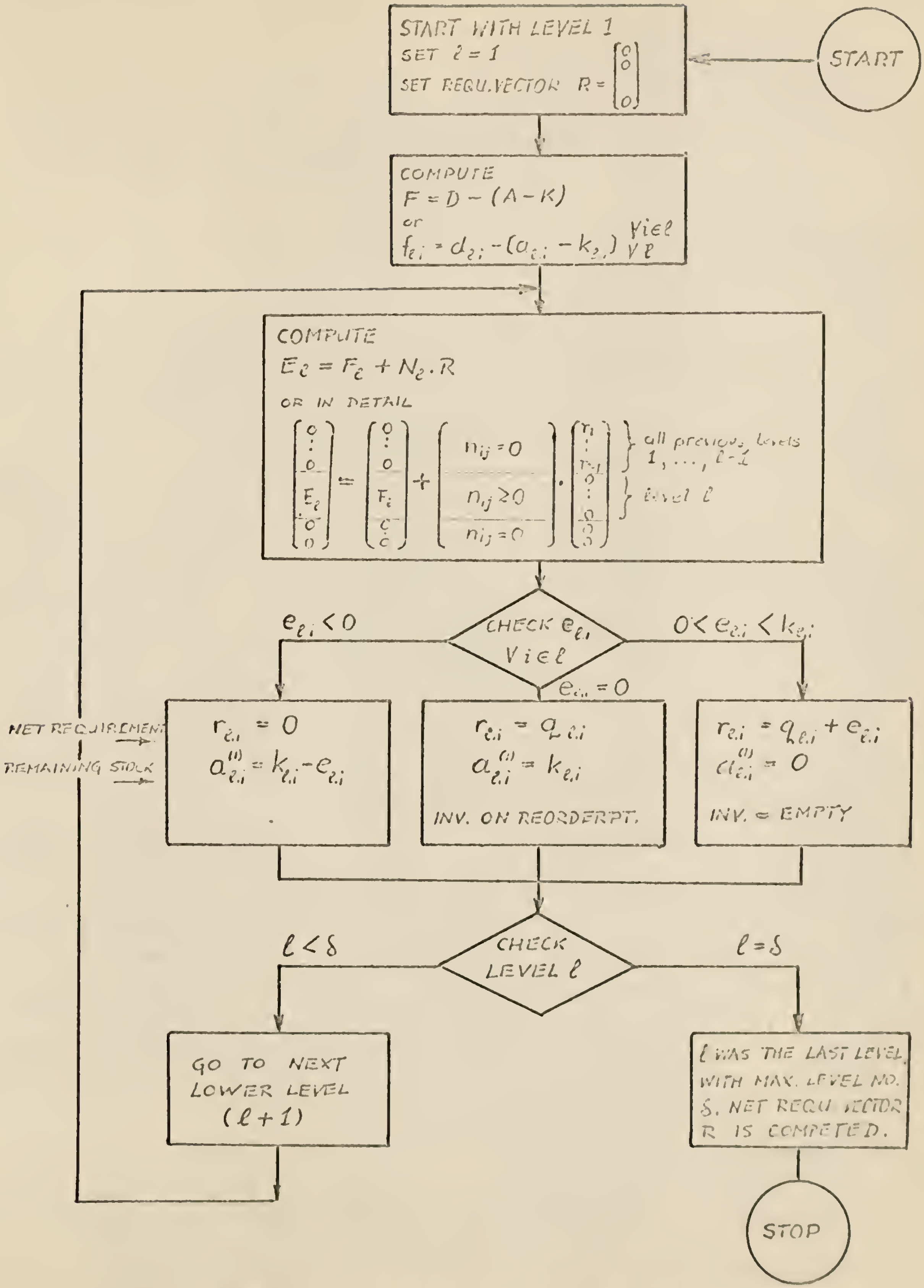


Figure 3.8. Blockdiagram for reorder requirements.

for an explanation to section 3.5 and Figure 3.7 for all the needed details.

3.6 TIME-DEPENDENT REQUIREMENTS [27]

The time-dependent requirement problem is a refinement of the simple requirement problem with inventory. It considers the makespan to manufacture each part i and determines the requirements for a demand which is given for several planning periods. The demand for a number of time periods t for all parts i is predetermined by management. Here, the question arises when should the production of part i be started, such that the demand can be satisfied at any given time t.

3.61 Definitions

Demand. The demand of the market, forecasted for a certain number of future planning periods is represented in a demand matrix D. Similar to the column vector D of simple demand, each column in the D-matrix stands for one planning period. The (it)th element d_{it} of D reflects the demand of part i in the planning period t. Time interval t in connection with the demand matrix D, will always be the "latest possible completion time interval".

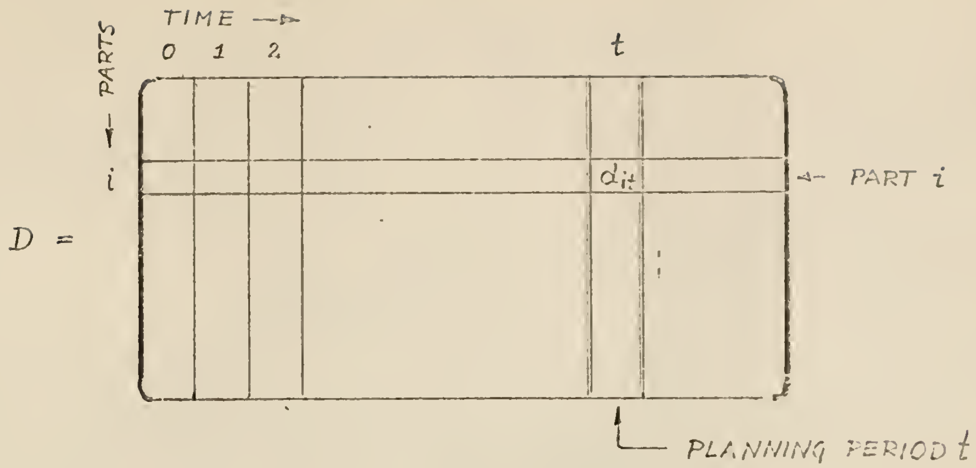


Figure 3.9. Demand matrix D.

Requirement. The time-dependent requirements are given in the requirement matrix R. The element $r_{\ell.i,t}$ represents the quantity of part $\ell.i$ required to start to be produced within the planning period t. The quantity $r_{\ell.i,t}$ has to fulfill the demand $d^{(c)}_{\ell.i,t+b_i}$ of part $\ell.i$ and the requirements $r^{(s)}_{k.i,t+b_i}$ for the parts of level 1 to $(\ell-1)$ of planning period $(t+b_{\ell.i})$. The letter $b_{\ell.i}$ stands for the manufacturing make span of part $\ell.i$. The time interval t used in connection with the requirement matrix R indicates the "latest possible starting time interval".

For completeness we define here the sub-matrix $R_\ell^{(s)}$ and $R_\ell^{(c)}$ of the net requirement matrix R. Similar to the N_ℓ-submatrix, both have the same dimensions (size) and the same elements as the R-matrix, except all elements not of level ℓ are zero elements. Sub-matrix $R_\ell^{(s)} \equiv R_\ell$ contains all net requirements of level ℓ to be started at the indicated time period t. The superscript "s" stands for "start". The sub-matrix $R_1^{(c)}$

contains all the net requirements of level ℓ to be completed at the indicated time period t. The letter "c" stands for "completion". Sub-matrix $R_\ell^{(s)} \equiv R_\ell$ is obtained from the $R_\ell^{(c)}$ matrix by individual left shifting of every element $r^{(c)}_{\ell.i,t}$ by $b_{\ell.i,t}$. The interpretation of the sub-matrices $G_\ell^{(s)}$ and $G_\ell^{(c)}$ is similar.

3.62 Elements of the D-R transformation

The problem to determine the requirements R from the given demand D can be seen as a transformation of the D-matrix into the R-matrix. This transformation is constrained by the following:

1. Direct demand D_ℓ
2. Indirect demand of lower-level-parts in higher-level-parts

$$\sum_{k=1}^{\ell-1} N_{\ell k} R_k$$

3. Netting process reduces the calculated gross requirements against an available inventory.
4. Set-back: The time difference between start and completion of production, where the completion time is given within the D and/or $R_\ell^{(c)}$-matrix, and the start time is computed in the R or the $R_\ell^{(s)}$-matrix.
5. Dependency of the time-difference $b_{\ell.i}$ from the production quantity $r^{(c)}_{\ell.i}$

$$b_{\ell.i} = f(r^{(c)}_{\ell.1})$$

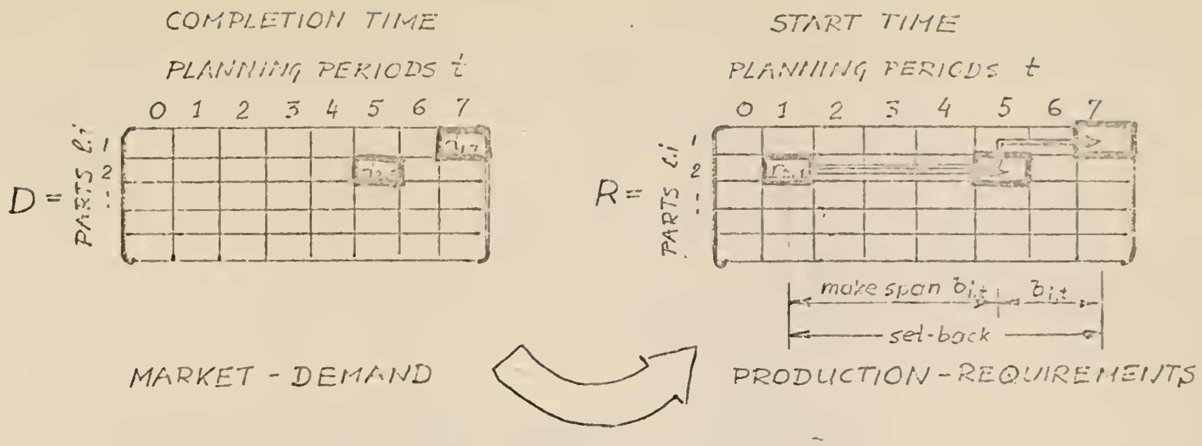


Figure 3.10. Transformation of D-matrix into R-matrix.

As with simple demand, in section 3.2, we compute here the time dependent requirements level by level, with increasing level number. Within each level the following steps are applied.

<u>3.62.1 Direct demand</u>

Direct demand was defined as the quantity of parts which are directly delivered to the customer. This quantity does not include the consumption of lower-level parts in higher-level parts. The direct demand can be read from the demand matrix D.

<u>3.62.2 Indirect demand</u>

The direct demand of part i in turn creates an "indirect" demand for sub-assemblies which are needed to assemble part i. To determine the demand for parts at any level, the complete structural information on each level in the manufacturing process is needed. This information is given by the N-matrix. For the

computation of the indirect demand we apply formula (3.6) of the simple demand.

$$\begin{aligned}
&\text{Level 1:} && I_1 = 0 \\
&\text{Level 2:} && I_2 = N_{21} \cdot R_1 \\
&\text{Level 3:} && I_3 = N_{31} \cdot R_1 + N_{32}\ R_2 \\
&\text{Level 4:} && I_4 = N_{41} \cdot R_1 + N_{42}\ R_2 + N_{43} \cdot R_3 \\
&\text{Level } \ell\text{:} && I_\ell = \sum_{k=1}^{\ell-1} N_{\ell k} \cdot R_k \qquad \forall\ \ell = 1, \ldots, \delta
\end{aligned} \tag{3.24}$$

For the calculation of the indirect demand of level-ℓ parts in level-k parts ($k = 1, 2, \ldots \ell - 1$) the netted and set-back requirement quantities of $R_k^{(s)}$ ($k = 1, 2, .. \ell - 1$) must be used. It is noted here, that the direct and indirect demand will be calculated together at any level ℓ, as done in section 3.4 with equation (3.6):

$$G_1^{(c)} = D_1^{(c)} \tag{3.15}$$

$$G^{(c)}{}_\ell = D^{(c)}{}_\ell + \sum_{k=1}^{\ell-1} N_{\ell k} \cdot R_k{}^{(s)} \qquad \forall\ \ell = 2, \ldots, \delta \tag{3.26}$$

3.62.3 Available inventory "on hand"

If there are parts available in stock, the requirements will be taken from this stock, i.e. netted against stock. The objective of this calculation is to use as many items as possible from the available stock. Whenever the requirements are netted against the available inventory, the requirements for parts at a lower level change. To reach the goal (to use as many items as

possible from the available stock) and to avoid a cumbersome recalculation of the requirements for parts of lower levels, the netting operation is started at the highest level, at level 1, going down level by level.

The adjusting of the requirements against the available inventory has to be performed just after the calculation of gross requirements $G_{\ell}^{(c)}$ for level ℓ has been completed. The gross requirements are reduced by the available inventory, starting with the earliest existing (or most urgent) requirements quantity, i.e,, the left most requirement quantity $g_{\ell.i,t}$, going step by step into the future periods $g_{\ell.i,t+1}$, $g_{\ell.i,t+2}$, ... and so on, till the inventory is exhausted. The netting operation of level ℓ is completed when all parts $\ell.i$ have been adjusted against the on hand stock.

We extend the meaning of the diminish symbol $\ominus$ in equation (3.23) and formulate the above stated algorithm with:

$$R_{\ell}^{(c)} = \{D_{\ell}^{(c)} + \sum_{k=1}^{\ell-1} N_{\ell k} R_k^{(s)}\} \underset{t}{\ominus} A_{\ell} \tag{3.27}$$

where the extended diminish symbol $\underset{t}{\ominus}$ stands for the time-depending procedure of this section. The block diagram in Figure 3.11 shows the main idea of the algorithm to generate the netted requirements with an on hand inventory.

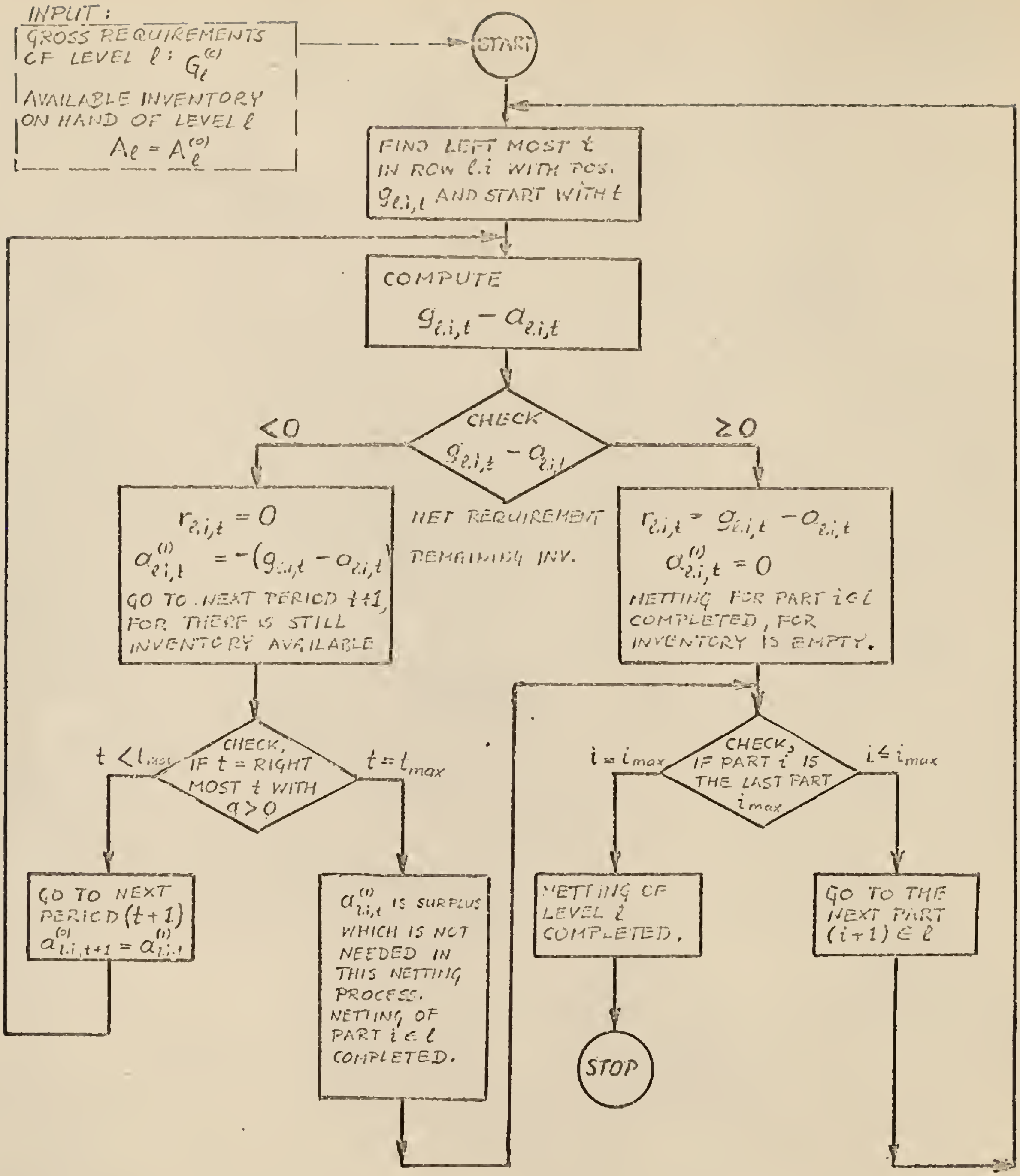


Figure 3.11. Blockdiagram for netting with "on hand" inventory.

3.62.4 Available inventory "on order"

In the previous section 3.62.3 we considered only parts which are completed and "on hand" in a regular stock. The unfinished parts in process and on order have been neglected. In the complete requirements situation all these parts have to be included in the computation. These parts are available in a certain future period. The parts in process (manufactured in the own plant) and on order (ordered detail parts from a vendor) are called "on order" quantities, in contrast to the "on hand" inventory.

The on order analysis supplies, for every undelivered order (internal or external) of a part i, the expected due date $\hat{t}$ and the quantity a_i. This information can efficiently be stored in an availability matrix A, where the (it)th element a_{it} represents the quantity of part i due to be available in the time period t. The due date $\hat{t}$ lies within the time period t. Conveniently, we define the first column t = 0 of the A-matrix as the "on hand" inventory, and is identical to the A-vector of section 3.62.3. The time period t = 0 represents the present planning period.

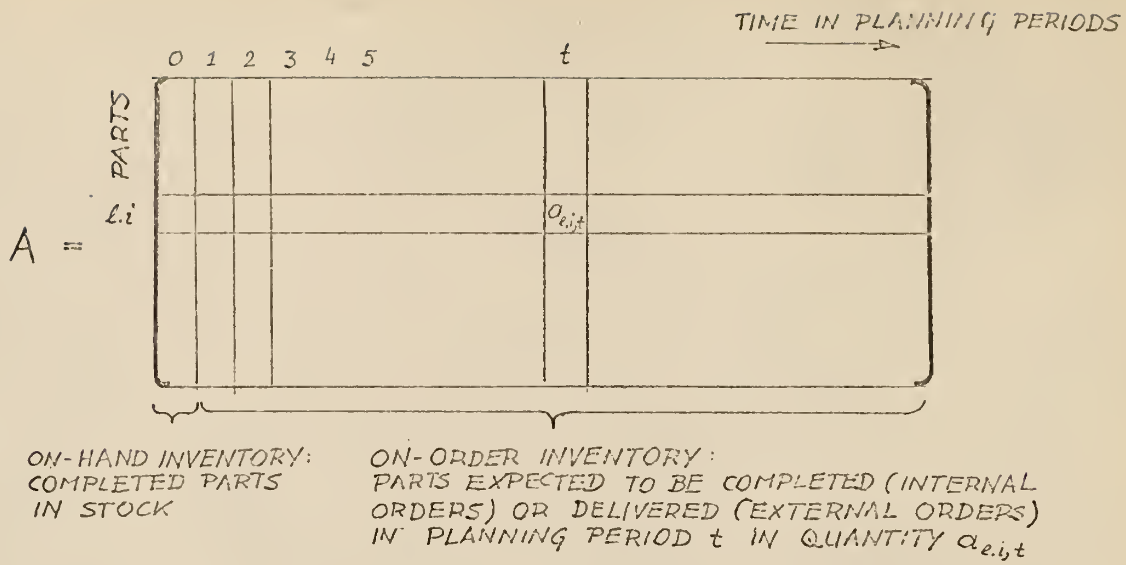


Figure 3.12. Availability-matrix A

The idea of the on order inventory changes the netting procedure. The gross requirements $g^{(c)}_{\ell.i,t}$ of period t will be adjusted against the surplus inventory $\sum_{t=0}^{t-1} a^{(1)}_{\ell.1,t}$ from period 0 up to and including (t - 1) plus the on order inventory of period t, $a^{(0)}_{\ell.i,t}$. This can be stated with the diminish symbol $\ominus$ of equation (3.23):

$$r^{(c)}_{\ell.i,t} = g^{(c)}_{\ell.i,t} \ominus \{\sum_{\tau=0}^{t-1} a^{(1)}_{\ell.i,\tau} + a^{(0)}_{\ell.i,t}\} \qquad (3.28)$$

or eliminating g:

$$r^{(c)}_{\ell.i,t} = \{d^{(c)}_{\ell.i,t} + \sum_{\substack{k=1 \\ \forall j \in k}}^{\ell-1} n_{\ell.i,k.j}\; r^{(s)}_{k.j,t}\} \ominus \{\sum_{\tau=0}^{t-1} a^{(1)}_{\ell.i,\tau} + a^{(0)}_{\ell.i,t}\} \qquad (3.29)$$

whereby the remaining of "surplus" inventory is given with

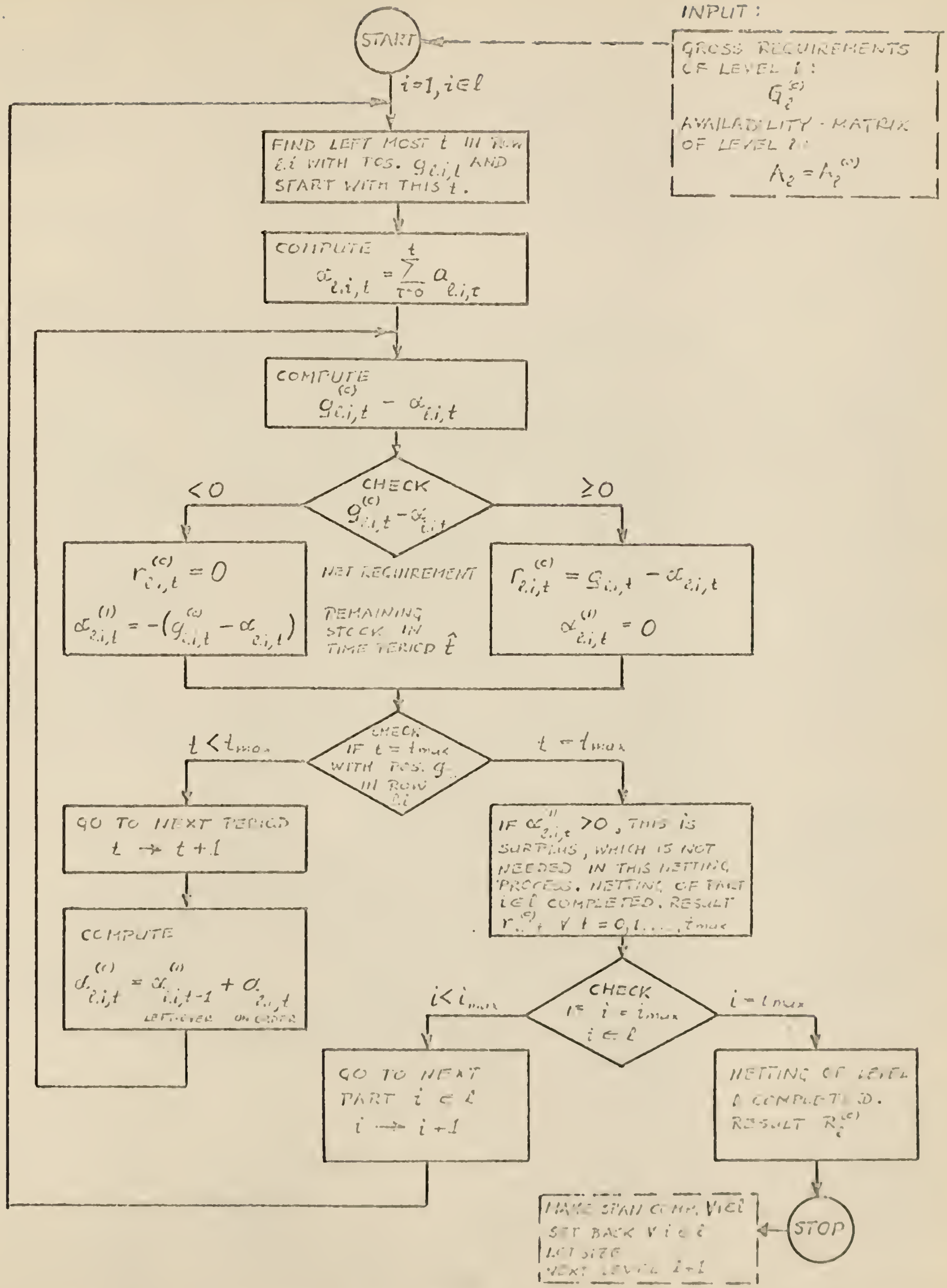


Figure 3.13. Block diagram for netting with on-hand and on-order inventory.

$$a^{(1)}_{\ell.i,t} = \{\sum_{\tau=0}^{t-1} a^{(1)}_{\ell.i,\tau} + a^{(0)}_{\ell i,t}\} \ominus \{g^{(c)}_{\ell i,t}\} \qquad (3.30)$$

The modified block diagram is displayed in Figure 3.13.

3.62.5 Set-back

The make span is the time difference between the start event and the completion event. It includes the operation periods of: paperwork preparation, set-up-time on the facility, run-time on the facility, and safty-time span. The completion event is given as the quantities $d_{i,t}$ of part i in the period t in the demand matrix D. The start time t has to be determined from the demand matrix D and the make span B to construct the desired requirement matrix R. The make span b_i is always represented as an integer multiple of the planning period unit we have chosen. The computation operation using the make span to determine the start date is called "set-back", for it shifts the necessary manufacturing operations from the future completion event back toward the current time period.

3.62.6 Quantity dependency of the set back

Case 1, quantity independent of make span. Assuming the make span is approximately equal for all quantities, within a certain range, the whole row ℓ.i of part ℓ.i is shifted by the

make span $b_{\ell.i}$ to the left, toward the current time period. For intermitant manufacturing with relatively constant lot size, generally all consumed parts have approximately the same set back, independent of the quantity to be produced. This is, of course, not true for flow production.

Case 2, quantity dependent upon make span. Diving into a more accurate but much more elaborate computation, we assume that the make-span is dependent of the quantity $r_{\ell.i,t}$ to be manufactured. In this case every quantity $r_{\ell.i,t}$ of part $\ell.i$ in the time period t has to be shifted individually for i and t to the left. A very good approximation of the make span is the assumption that it is a linear function of the required quantity or the manufactured quantity $r^{(c)}_{\ell.i,t}$. We state:

$$b_{\ell.i} = u_{\ell.i} + r^{(c)}_{\ell.i}\,\tau_{\ell.i} \tag{3.32}$$

and determine with

$b_{\ell.i}$ = the make span or lead time in time units,

$u_{\ell.i}$ = the quantity independent constant; this time period includes the time for preparation, set-up on the facilities, and a safty-time-span.

$\tau_{\ell.i}$ = the run time of one part i of level ℓ on the facilities,

$\ell.i$ = the index $\ell.i$ indicates the ith part, arranged in level ℓ.

Starting with the smallest t, where a positive $d_{\ell.i,t}$ or $r^{(c)}_{\ell.i,t}$ shows up all $d_{\ell.i,t}$ or $r^{(c)}_{\ell.i,t}$ are shifted individually for every part $\ell.i$ and time t to the left, from t to $(t-b_{\ell.i})$, for the make span $b_{\ell.i,t}$ is a function of $r^{(c)}_{\ell.i,t}$ itself.

In the case that a left shifted $r^{(c)}_{\ell.i,t_1}$ from the period t_1 meets with another $r^{(c)}_{\ell.i,t_2}$ from the time period t_2 in the same time period t_0, $r^{(c)}_{\ell.i,t_1}$ and $r^{(c)}_{\ell.i,t_2}$ are not added together. This event uses the following concept: For $t_1 < t_2$ the quantity $r^{(c)}_{\ell.i,t_1}$ will be shifted additionally to the left by $b_{\ell.i,t_1}$ to the time period $(t_1 - 2 \cdot b_{\ell.i,t_1})$. The reason for this concept is the assumed quantity dependent make span. To avoid an over-load of the production, we start to produce this lot earlier. See Figure 3.14.

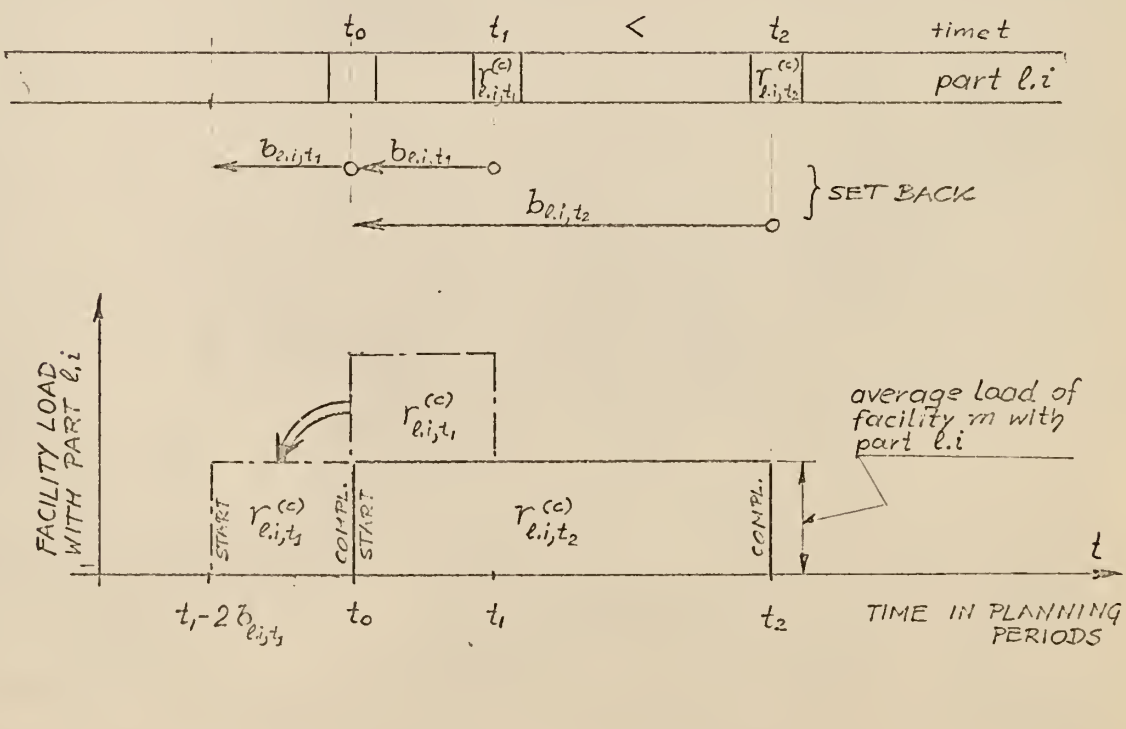


Figure 3.14. Set back and facility load.

3.63 The D-R transformation

After describing all the main elements of the transformation separately, we consider now the total computation of the time-dependent requirement. We will explain the procedure here in detail for any level ℓ.

We start with the final assemblies, with level number 1, to make the computation of the indirect and of the netted requirement as economic as possible.

We consider level ℓ. For each production period t the sum of the direct demand D_ℓ plus the indirect demand for higher level parts $\sum_{k=1}^{\ell-1} N_{\ell k} \cdot R_k^{(s)}$ is calculated. Remember that for the calculation of the indirect demand of "level-ℓ" parts in higher level parts ($k = 1, 2, .. \ell - 1$), the netted, and set-back quantities of $R_k^{(s)}$ are used. An inventory reduces the gross requirements $G_\ell^{(c)} = D_\ell^{(c)} + \sum_{k=1}^{\ell-1} N_{\ell k} \cdot R_k^{(s)}$ to the net requirements $R_\ell^{(c)}$ the quantities necessary to be manufacture or to be ordered. The net requirements $R_\ell^{(c)}$ are determined according to the algorithm of section 3.62.4. With the diminish symbol Θ of equation (3.23) we state:

$$r^{(c)}_{\ell.i,t} = \{d^{(c)}_{\ell.i,t} + \sum_{\substack{k=1 \\ \forall j \varepsilon k}}^{\ell-1} n_{\ell.i,k.j} \cdot r^{(s)}_{k.j,t}\} \Theta \{\sum_{t=0}^{t-1} a^{(1)}_{\ell.i,t} + a^{(0)}_{\ell.i,t}\} \qquad (3.33)$$

The time period t, in which the net requirements $r^{(c)}_{\ell.i,t}$ stand, represent, for this lot, the latest possible completion time.

To manufacture item $\ell.i$, requires at least $b_{\ell.i}$ planning periods, the make span. Hence, all parts $\ell.i$ are shifted individually $b_{\ell.i}$ units to the left, i.e. closer to the current planning period $t = 0$, from the period t to $(t - b_{\ell.i})$. With this set-back procedure we develop the sub-matrix $R_{\ell}^{(s)}$.

$$R_{\ell}^{(s)} = \text{SET-BACK}\ (R_{\ell}^{(c)}) \tag{3.34}$$

In the case the set-back is to be assumed quantity dependent, the make span must first be calculated individually for every part $\ell.i$ and time t, according to the rules in section 3.62.6 and then the set-back operation performed.

With the set-back operation level ℓ has been completed, and the same steps are now applied on the next lower level $(\ell+1)$. The computation procedure for the requirement quantities, the netting operation, the make span, the set-back operation and the lotting is shown in Figure 3.15.

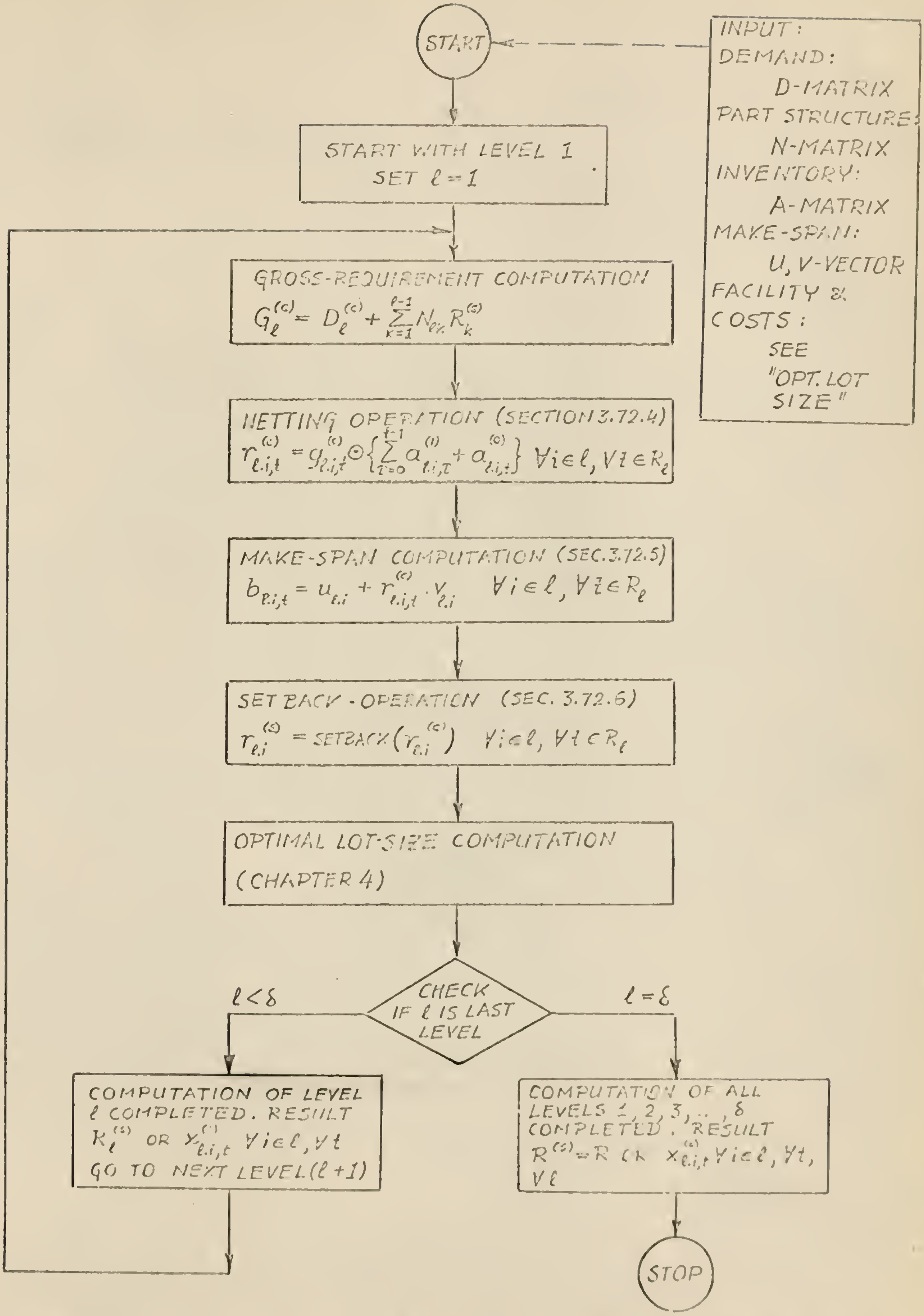


Figure 3.15. Blockdiagram for computation of net-requirement R in optimal lot sizes $x_{\ell.i}$

4. OPTIMAL LOT SIZE

A company produces many parts to meet a given market demand D for several future periods. The production has to fulfill the requirements R for each part. The available production hours of each machine, the time it takes to manufacture each required part, the cost for each set-up and the costs of carrying an inventory are known. The problem is to determine the lot size for each individual part so that the production requirements are met and the combined costs of manufacturing and inventory are minimized.

4.1 EXAMPLE [41]

The optimal lot size problem is illustrated with a simple example. The requirements of a single part are described by the cumulated requirement function; see Figure 4.1. The horizontal axis represents the planning periods in months, the vertical axis shows the cumulative requirements which have been calculated with the algorithm of Chapter 3.

At the end of February there is, according to Figure 4.1, an accumulated requirement for 30 parts, by the end of March there is an accumulated requirement of 40 parts, by the end of April 65 parts, and so on. The graph shows two possible production plans to meet the given requirements. The first plan specifies that 80 parts are made in January and 40 parts in

April. All requirements are met with two set-ups for the total planning period of six months. The <u>second plan</u> indicates a production of 30 parts in January, 10 parts in February, 25 parts in March, 35 parts in April, and 20 parts in May. The requirements will be met with 5 set-ups in the total planning period of six months. Both of the production plans meet the given requirements. Plan 1 requires two set-ups, whereas plan 2 involves five set-ups. Each set-up creates costs, not only the setting up of the machines in the workshop, but also the necessary paper work. It is clear that the first plan generates less set-up costs than plan 2. On the other hand each part produced for a latter period of usage has to be stored in an inventory.

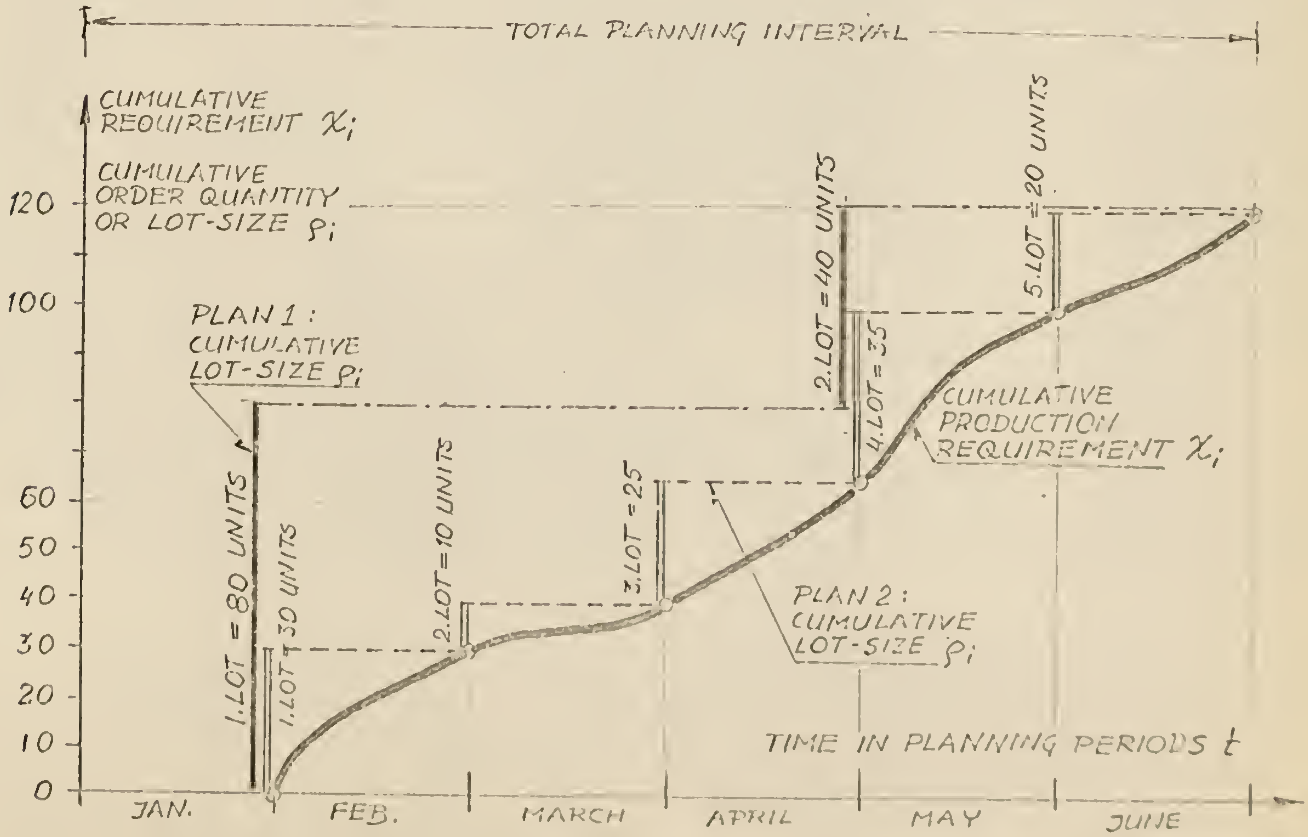


Figure 4.1. Market demand and production plans.

Each inventory has to be maintained and, hence, generates costs. Plan 1 requires a larger inventory and creates more carrying costs than plan 2.

Considering both, inventory and set-up costs, which of the two plans represents the pattern with the lower costs? Furthermore, does either of these two plans represent the most economic plan?

4.2 OPTIMAL LOT SIZE MODEL WITH NON-CONSTANT DEMAND AND SEVERAL PARTS

Let us consider a manufacturing company that produces many parts to meet a certain given set of requirements. The plan for the future production periods has to be established and the economic lot size for every part has to be determined. The requirements of each part fluctuates from one period to another. The following data are known:

- the normal available production time η_{mt} of each machine (facility) m in each production period t. The "normal" machine capacity η_{mt} includes only the time for the normal production level;
- the maximal available production time ϕ_{mt} of each machine m in each production period t; the "maximal" machine capacity ϕ_{mt} includes the normal available production time η_{mt} and the overtime. In mathematical terms we write: $\eta_{mt} \leq \phi_{mt}$ and call $(\phi_{mt} - \eta_{mt})$ the maximal available overtime of period t on machine m.

- the manufacturing time (run-time) for each single part i on a certain machine m, τ_{im}
- the set-up time for each part i on the specific machine m, σ_{im}
- the cost of each set-up, for part i on machine m, c_{im}, and
- the cost of carrying a part i in the inventory, pc_i, where p denotes the carrying costs expressed as a percentage of the value of the stored item c_i for the total planning period.

The problem is to determine the order quantities or the lot size x for each individual part, so that the requirements are met and the combined costs of set-up, inventory and overtime minimized over all parts.

The formulation of the lot-size problem includes as a special case the conventional economic lot-size formula. An abundance of facilities, that is machines and labor, enables one to plan for each part separately. The additional restriction, demand per time period is constant, leads to the conventional economic lot size formula.

4.21 Mathematical statement of the problem [41]

4.21.1 Requirement (or market demand) restrictions

The manufacturing company produces I parts i on the M available machines m. We are asked to plan for the T future planning periods t. The requirements r_{it} of each part i and each planning

period t are given. We must determine the quantity x_{it} of part i that is to be manufactured in the period t, and call x_{it} the lot size or order quantity. We shall use the cumulative requirement and the cumulative order quantities to develop the mathematical formulation of the problem. The cumulative requirements for the part i from t = 0 up to and including period t are denoted by ρ_{it}, the cumulative order quantities for part i are expressed with χ_{it}. In mathematical terms we write:

$$\rho_{it} = \sum_{k=0}^{t} r_{ik} \tag{4.1}$$

$$\chi_{it} = \sum_{k=0}^{t} x_{ik} \tag{4.2}$$

The production plan must meet the given requirements. This means that the cumulative order quantities must be larger or not less than the cumulative requirements. This restriction can be expressed by the inequality:

$$\chi_{it} \geq \rho_{it} \tag{4.3}$$

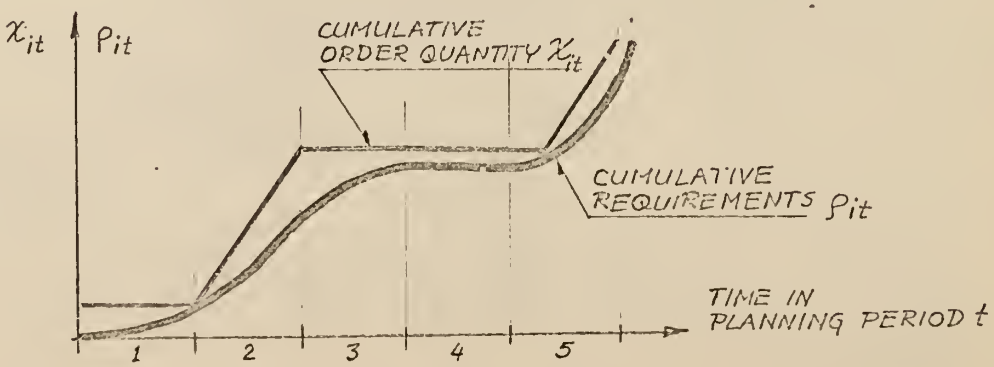


Figure 4.4. Relation between requirements and production.

4.21.2 Machine restrictions

The production plan must stay within the available machine capacities ϕ_{mt}. To satisfy this restriction we need to determine the machine time required by the production plan. The production of a single part i on machine m needs τ_{im} hours. To manufacture part i on the machine m the machine has to be prepared for this production and we denote with σ_{im} the set-up time. Compute how large a load is imposed on machine m producing a lot of part i. The quantity x_{it} requires $\tau_{im}x_{it} + \sigma_{im}$ hours to be manufactured on machine m. If we do not fabricate any part in this period, $x_{it} = 0$. The previous formula would tell us that the machine load for $x_{it} = 0$ is σ_{im}, which is incorrect. There is no need for setting up the machine, if we do not manufacture any single part. Consequently we need a qualifying statement: Machine load = 0 if $x_{it} = 0$. The machine load of machine m by part i is expressed with:

$$\text{machine load} = \begin{cases} \tau_{im}x_{it} + \sigma_{im} \;\ldots & \text{if } x_{it} > 0 \\ 0 \;\ldots & \text{if } x_{it} = 0 \end{cases}$$

To put these two lines into a single statement, we introduce a switch or "unit function u(x)". The unit function u(x) is defined as:

$$u(x) = \begin{cases} 1 \text{ for } x > 0 \\ 0 \text{ for } x = 0 \end{cases} \tag{4.4}$$

The graph of the unit function u(x) is shown in Figure 4.5.

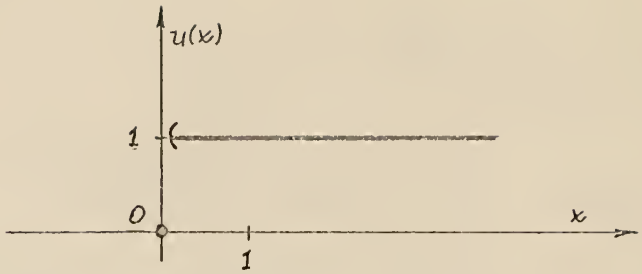


Figure 4.5. Unit function u(x).

With the introduction of the unit function u(x) we simplify the statement of the machine load and write:

$$\text{machine load} = \{\tau_{im} x_{it} + \sigma_{im} \cdot u(x_{it})\} \ .$$

For the total required time, h_{mt}, on machine m in the planning period t, all parts i processed on the machine m, must be considered.

$$h_{mt} = \sum_i \tau_{im} x_{it} + \sum_i \sigma_{im} u(x_{it}) \qquad (4.5)$$

The machine load h_{mt} associated with our production plan must stay within the available machine capacity ϕ_{mt}. The following inequality must hold:

$$\text{machine load} \leq \text{machine capacity}$$

$$h_{mt} \leq \phi_{mt}$$

$$\sum_i \tau_{im} x_{it} + \sum_i \sigma_{im} u(x_{it}) \leq \phi_{mt} \qquad (4.6)$$

Equation (4.3) assures that the production plan meets the requirements (and with this the market demand) and equation (4.5) that the production plan is within the available machine capacity. A production plan fulfilling these two necessary conditions is a _feasible_ production plan. The feasible plan is not necessarily the _optimal_ production plan, i.e. the plan generating the lowest costs.

4.21.3 Inventory cost

The production plan specifies that a cumulative quantity of χ_{it} parts is manufactured in the period t, where as there are only ρ_{it} parts required. All these $(\chi_{it}-\rho_{it})$ extra parts produced in period t for a later usage have to be stored, what requires maintenance of an inventory, which in turn creates costs. To carry an inventory of $(\chi_{it}-\rho_{it})$ units of part i in the period t generates $(\chi_{it}-\rho_{it})pc_i$ inventory costs in period t. We call:

c_i - the cost of each part i,

pc_i - the carrying cost of each part i for one period t, and

C_I - the total inventory cost.

The total inventory carrying costs are obtained by adding the inventory cost of each individual part i for each period t. We state:

$$C_I = \sum_i \sum_t pc_i \; (\chi_{it} - \rho_{it})$$

$$C_I = p.\sum_i \sum_t c_i (\chi_{it} - \rho_{it}) \qquad (4.7)$$

4.21.4 Set-up cost

Set-up cost refers to the cost of changing over the production process to manufacture the required part i. Every set up of a machine for a different lot generates costs. We denote with:

$\hat{c}_{im}$ = the set-up costs of manufacturing part i on machine m, and

C_s = the total set-up costs, which are calculated with:

$$C_s = \sum_i \sum_m \sum_t \hat{c}_{im} . u(x_{it}) \qquad (4.8)$$

We note, there is not more than one set-up per planning period t and per machine m for a specific part i.

4.21.5 Overtime cost

The required machine load h_{mt} must stay within the "maximal" machine capacity ϕ_{mt}, as stated in section 4.21.2. The production

time outside the normal machine capacity $\acute{\eta}_{mt}$ is punished with additional costs, the overtime costs. The overtime $\{h_{mt} \ominus \eta_{mt}\}$ on machine m generates in period t overtime costs of $\{h_{mt} \ominus \eta_{mt}\}.\bar{c}_m.\bar{p}$, where:

$\bar{c}_m$ = the normal production costs per hour of machine m, and

$\bar{p}$ = the overtime costs expressed as a percentage of normal production costs per hour $\bar{c}_m$.

The total overtime costs C_0 for all machines m and all time periods t is given with:

$$C_0 = \sum_m \sum_t \{h_{mt} \ominus \eta_{mt}\} \bar{c}_m \bar{p} \tag{4.9}$$

$$= \bar{p} \sum_m \sum_t \{[\sum_i \tau_{im} x_{it} + \sum_i \sigma_{im} .u(x_{it})] \ominus \eta_{mt}\} \bar{c}_m \tag{4.10}$$

4.21.6 Total cost

Adding up inventory, set-up and overtime costs we obtain the total costs for our production plan:

$$C_{total} = C_I + C_S + C_0$$

$$= p \sum_i \sum_t c_i (\chi_{it} - \rho_{it}) + \sum_i \sum_m \sum_t \hat{c}_{im} u(x_{it}) +$$

$$+ \bar{p} \sum_m \sum_t [\sum_i \tau_{im} x_{it} + \sum_i \sigma_{im} .u(x_{it})] \ominus \eta_{mt}\} \bar{c}_m \tag{4.11}$$

4.22 Summary of the mathematical statement

1.) $\chi_{it} \geq \rho_{it}$ $\quad \forall t$ and $\forall i$ (4.12)

with

$$\rho_{it} = \sum_{k=0}^{t} r_{ik} \quad \forall t \text{ and } \forall i \tag{4.13}$$

$$\chi_{it} = \sum_{k=0}^{t} x_{ik} \quad \forall t \text{ and } \forall i \tag{4.14}$$

2.) $h_{mt} \leq \phi_{mt}$ $\quad \forall t$ and $\forall m$ (4.15)

with

$$h_{mt} = \sum_{i} \tau_{im} . x_{it} + \sum_{i} \sigma_{im} u_{it} \tag{4.16}$$

3.) $$u_{it} = \begin{cases} 0 \text{ if } x_{it} = 0 \\ 1 \text{ if } x_{it} > 0 \end{cases} \tag{4.17}$$

Objective function:

$$C_{total} = p\sum_{i} \sum_{t} c_i (\chi_{it} - \rho_{it}) + \sum_{i} \sum_{m} \sum_{t} \hat{c}_{im} . u_{it} + \bar{p} \sum_{m} \sum_{t} \{h_{mt} \ominus \eta_{mt}\} \bar{c}_m \tag{4.18}$$

with

$$(h_{mt} \ominus \eta_{mt}) = \begin{cases} (h_{mt} - \eta_{mt}) & \text{if } h_{mt} > \eta_{mt} \\ 0 & \text{if } h_{mt} \leq \eta_{mt} \end{cases} \tag{4.19}$$

The problem is to acquire the set of x_{it} that we minimize the total cost equation (4.18) subject to the inequalities (4.12) and (4.15). This optimization problem is a special case of a linear programming problem, for it includes the unit or indicator function u(x), and can be solved with a "mixed-integer-linear programming" method. For the algorithm to solve this problem we refer to R. Gomory: "An algorithm for the mixed integer problem", Rand Report P - 1885, June 23, 1960; and G. Dantzig: "On the significance of solving linear programming problems with some integer variables", Econometrica, Vol. 28, 1, Jan. 1960.

4.23 Remarks

To realize the magnitude of the problem, we consider a realistic example. Assume we produce I = 500 parts i on M = 50 machines m and our total planning interval is T = 10 planning periods t.

Unknown. In our system we have I = 500 unknowns in each of the 10 planning periods, that is a total of I.T = 500 . 10 = 5000 unknowns in the total system. From the practical viewpoint, the increasing of the number of unknown x_{it} with the number of planning periods (for a fixed total planning interval) results in a more accurate computation and control of the total cost.

Equations. Equation (4.14) relates the cumulative order quantities χ_{it} to the order quantities x_{it}. Equation (4.14) has to be written for each part i and each production period t. This means, equation (4.14) represents I.T = 500 . 10 = 5000 equations. Similar equation (4.12) represents once more I.T. = 500 . 10 = 5000 equations. The condition for staying within the maximal machine capacity ϕ_{mt} is given by equation (4.15) and is valid for each of the M machines m and each of the T production periods t. Equation (4.15) stands for M.T. = 50 . 10 = 500 inequalities. The total cost equation (4.18) contains of three sums. The first sum contains I.T = 500 . 10 = 5000, the second sum I.M.T. = 500 . 50 . 10 = 25,000, and the third sum M.T = 50 . 10 = 500 terms. We gain a total of 30,500 terms.

In summary we have 5,000 unknowns, 10,500 inequalities and an objective function with 30,500 terms. These large numbers result from the interaction of all parts i, however, in practice, many of the enumerated terms will be zero and thus reduce the predicted theoretical magnitude of the problem.

5. SCHEDULING FOR JOB AND LOT PRODUCTION

The problem of scheduling of job-shop-type production allows no precise mathematical solution when viewed from an economical viewpoint. We show in this chapter, that the method of assembly-flow-scheduling can be extended to include job-shop-type scheduling, if we do not insist on an exact solution for this problem.

5.1 SCHEDULING WITH THE GANTT CHART

One of the most common tools used to plan a schedule is the Gantt chart. This is a diagram representation, showing the start and completion time of either the successive articles which are processed by each machine or the successive machine which process each article. The idea of this chart is simple. The horizontal axis represents the time axis, divided into planning periods of suitable length. Each line in the chart represents a certain machine group, labor class or article. The article Gantt chart shows the successive facilities or machines which process each article or part. Each row represents a certain article. The facility Gantt chart, which is much more common, displays the successive articles or parts which are processed by each of the involved facilities. Every line stands for a certain facility, machine or labor group.

Example [40]. We shall illustrate the idea of the Gantt chart with an example, see Figure 5.2. The horizontal axis describes time in manufacturing periods. On the vertical axis, four different facility groups are indicated, each refering to a particular machine or machine group.

Operation sheet of Part A_1:

Operation No.	Facility No.	Estimated Time (units)
1	1	5
2	3	7
3	2	9

Operation sheet of Part A_2:

Operation No.	Facility No.	Estimated Time (units)
1	3	6
2	4	11
3	1	4

Operation sheet of Part A_3:

Operation No.	Facility No.	Estimated Time (units)
1	4	6
2	3	7
3	2	10

Figure 5.1. Operation sheets of part A_1, A_2, and A_3.

Part A_1 must, according to the operation sheet, Figure 5.1, be manufactured on facility 1, then 3 and finally on facility 2. The third lot of this part A_1 is shown in the Gantt chart and has been scheduled according to the operation sheet and the available time on the facilities.

Part A_2 runs through tree operations in the sequence of facilities 3, 4 and 1. The scheduling of this part creates no difficulty. We have to schedule the start of the first operation just after the completion of the first part A_1. Between each of the operations there is a delay or waiting time, until the next operation can be started.

For Part A_3 the situation is different. The operation sheet requires the facility sequence of 4, 3, and 2. Every time the lot is finished on one of the machines (machine 4 or 3) the lot has to wait a certain time for the following operation, since the machine required is occupied by another lot of part A_1 or A_2.

As long as all the lots are finished at the required completion day, there is no reason to change this pattern. In many manufacturing companies it is customary to prepare charts in advance similar to the one shown in Figure 5.2. They are used for any kind of planning purposes. One begins by setting up the shipping dates and then works "backwards". So a complete chart for the future is prepared. Every time a certain operation is completed, the foreman consults the chart and determines what

specific lot is to be taken on the machine. In case the lot is not available he knows the lot is late and he takes corrective actions.

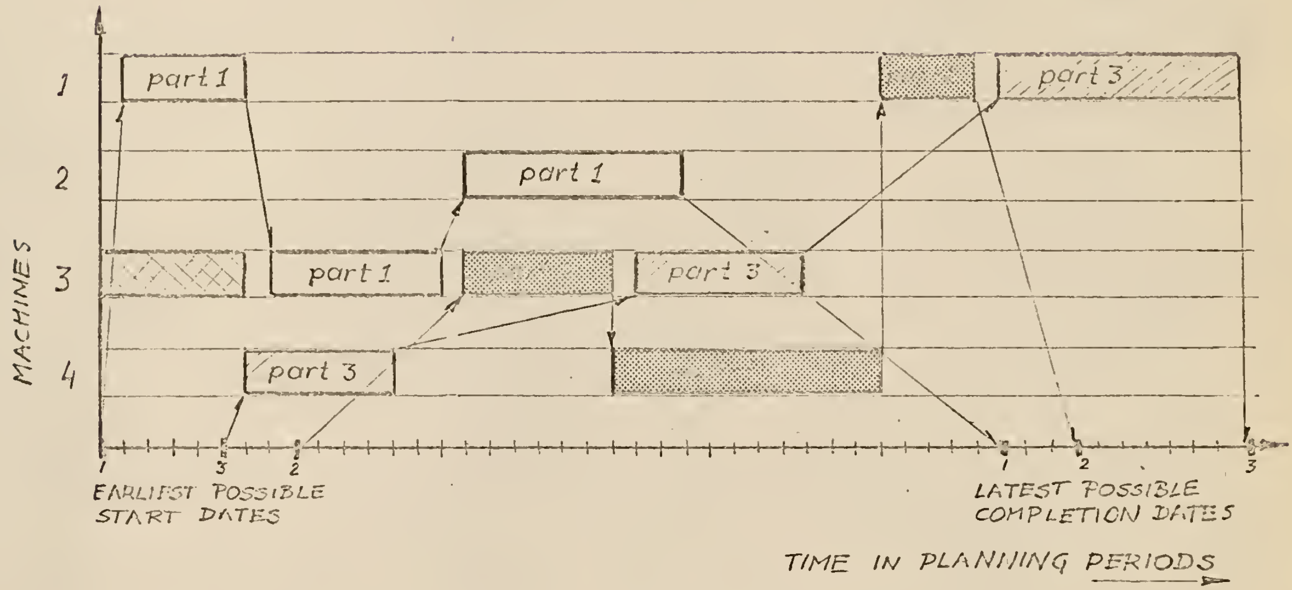


Figure 5.2. Gantt chart.

Changes owing to delayed lots [40]. Suppose we look at the Gantt chart this morning and try to determine what operations should be performed. We observe that we were on schedule up to today. The three indicated operations according to the Gantt chart were performed, as shown in Figure 5.2. At this moment we receive the information, that the next lot of part A_2 has not yet arrived and hence the first operation on the lot of A_2 cannot be performed.

Knowing, that this lot of part A_2 will not arrive in the near future we change the plan; we perform the second operation of the lot of part A_3 on facility 3. We perform also operation $L_{1,3}$ according to the original plan. After the lot $L_{3,2}$ is completed the delayed lot of A_2 arrives and consequently the first operation $L_{2,1}$ can be done on machine 3.

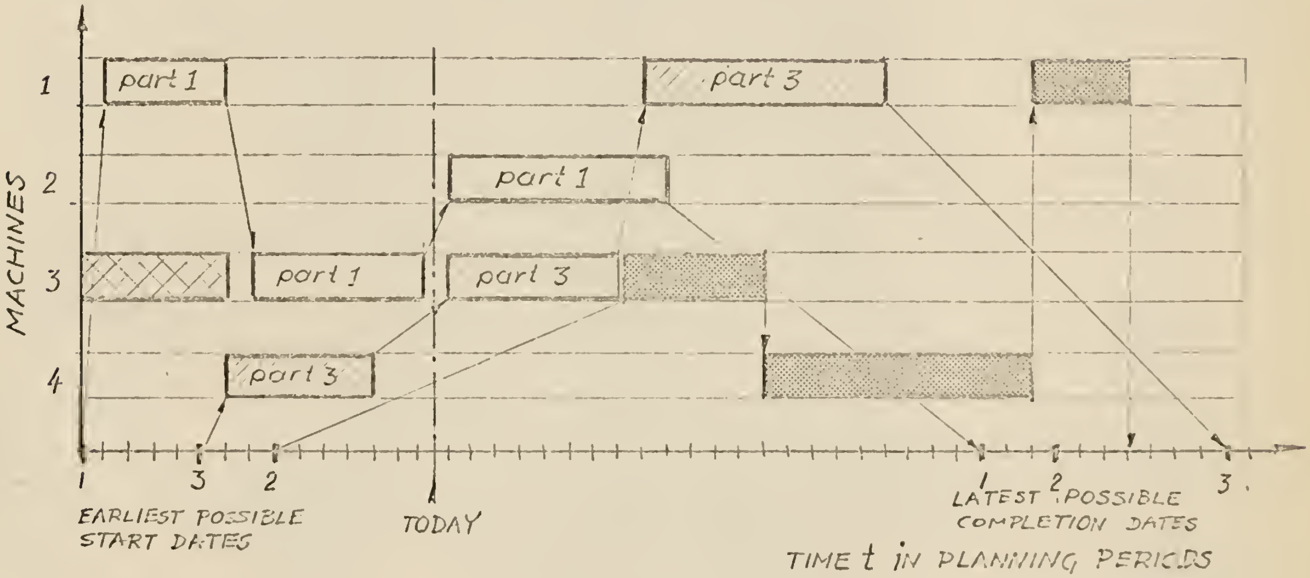


Figure 5.3. Changed Gantt Chart.

We recognize at this point that the original chart is useless because the situation has changed so much, we have to develop a new Gantt chart (see Figure 5.3). The event, that a certain lot is not available when needed may happen every day: machines break down, tools are not ready and available, labor is not available, and so on. Consequently this means that every day a complete new detail chart has to be developed. In a

company where hundreds of shippable parts are involved and thousands of items are manufactured this is almost an impossible job. On the other hand does it make sense to compute the schedule for two months in advance, knowing that every day we will have to rework the total schedule completely?

5.2 MATRIX REPRESENTATION OF THE GANTT CHART [12], [13], [17]

The graphical representation of the production schedule with the Gantt chart can be easily transformed into a matrix. The matrix use for the Gantt chart enables us to apply the advantages of the modern electronic computers.

5.21 Sequence matrix [12], [13]

The order in which a certain commodity c is processed by the different facilities f and the sequence in which a certain facility f accepts the commodities is the starting information for the development of the Gantt chart matrix.

The ith facility sequence, F_i, for each $i = 1,\ldots,m$, is the ordered set of indices of the successive facilities that process and produce the ith commodity c_i. Similar, the jth commodity sequence, C_j, for each $j = 1,\ldots,n$, is the ordered set of successive commodities that are processed by the facility f_j.

In general the facility sequence, F_i, is determined by the technological requirements to build a certain commodity and fixed on the operation sheet. The operation sheet specifies the type of facility and the sequence in which the facilities are to be used. In same cases the sequences may be open, leaves some freedom for scheduling, and can be specified later with the scheduling requirements. The commodity sequence, C_j, will be determined from the facility sequence, F_i, or from a given technical requirement.

The <u>facility sequence matrix</u>, F, is defined with the ordered collection of all the facility sequences.

$$F = \begin{array}{c} c_1 \\ c_2 \\ \vdots \\ c_m \end{array} \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{bmatrix} \begin{array}{l} \leftarrow \text{commodity } c_1 \\ \leftarrow \text{commodity } c_2 \\ \\ \\ \end{array} \tag{5.1}$$

Similarly, the <u>commodity sequence matrix</u>, C, is defined as the ordered collection of the commodity sequences.

$$C = \begin{array}{c} f_1 \\ f_2 \\ \vdots \\ f_n \end{array} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix} \begin{array}{l} \leftarrow \text{commodity sequence of facility } f_1 \\ \\ \\ \\ \end{array} \tag{5.2}$$

5.22 Quantitative and non-quantitative schedules

A schedule is non-quantitative if we assume that all operations have the same processing time $b_{ij} = 1$. Hence, a schedule is quantitative, if we imply the processing time may not be the same for all operations.

5.23 Non-quantitative Gantt chart matrix

The facility and commodity sequence matrix, when augmented by having idle times inserted at the proper places, will be called Gantt chart matrices, and are indicated by G(F) and G(C) respectively.

Example. Suppose we manufacture 3 commodities on 5 different facilities, that means $m = 3$ and $n = 5$. The facility sequence matrix, transfered from a hypothetical operation sheet, may be:

$$F = \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array} \begin{pmatrix} 2 & 1 & 2 & 4 \\ 1 & 5 & - & - \\ 2 & 1 & 3 & 5 \end{pmatrix} \qquad (5.3)$$

The facility sequence matrix F indicates, that the commodity c_1 is processes first on machine 2, than on machine 1, once more on machine 2 and finally finished on machine 4. Commodity c_2 is only processed on two facilities: machine f_1 and f_5. The third row shows the facility sequence of the commodity c_3: facility f_2, facility f_1, facility f_3 and finally facility f_5.

One possible commodity sequence matrix is:

$$C = \begin{array}{c} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{array} \begin{pmatrix} 2 & 1 & 3 \\ 1 & 3 & 1 \\ 3 & - & - \\ 1 & - & - \\ 2 & 3 & - \end{pmatrix} \qquad (5.4)$$

This sequence matrix indicates that facility f_1 processes the commodities in the order 2, 1 and 3. Another possibility would be the commodity sequence 2, 3, 1. This presentation is due to the same facility sequence of article c_1 and c_2 for the first two facilities: 2, 1. (see F-matrix).

$$C = \begin{array}{c} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{array} \begin{pmatrix} 2 & 1 & 3 \\ 3 & 1 & 1 \\ 3 & - & - \\ 1 & - & - \\ 2 & 3 & - \end{pmatrix} \quad C = \begin{array}{c} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{array} \begin{pmatrix} 2 & 3 & 1 \\ 3 & 1 & 1 \\ 3 & - & - \\ 1 & - & - \\ 2 & 3 & - \end{pmatrix} \quad C = \begin{array}{c} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{array} \begin{pmatrix} 2 & 3 & 1 \\ 1 & 3 & 1 \\ 3 & - & - \\ 1 & - & - \\ 2 & 3 & - \end{pmatrix}$$

The sequence matrix C and F together can be used to construct a feasable schedule, a Gantt chart matrix as follows:

$$G(F) = \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array} \begin{pmatrix} 2 & 1 & 2 & 4 & - \\ 1 & 5 & - & - & - \\ - & 2 & 1 & 3 & 5 \end{pmatrix} \qquad (5.5)$$

$$G(C) = \begin{array}{c} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{array} \begin{pmatrix} 2 & 1 & 3 & - & - \\ 1 & 3 & 1 & - & - \\ - & - & - & 3 & - \\ - & - & - & 1 & - \\ - & 2 & - & - & 3 \end{pmatrix} \qquad (5.6)$$

The Gantt chart is represented by the matrices G(F) and G(C). The matrix G(F) shows the successive facilities which process each part. The matrix G(C) shows the successive articles that are processed by each facility.

The (i,k)th element of the matrix G(F) is either a dash or an index of a facility. A dash means the ith part is not processed during the kth scheduling step. An index number of the jth facility shows that the article is processed by the jth facility during the kth step. The G(F) matrix has as many rows as there are parts.

The (j,k)th element of the matrix G(C) is either a dash or an index of a part. A dash represents that the jth facility is idle during the kth scheduling step. The index of the ith article means that the jth facility is processing the ith commodity during the kth scheduling step. The G(C) matrix has as many rows as there are facilities.

Each of the G(F) matrices can be rearranged into a G(C) matrix and vice versa. The matrices G(F) and G(C) which are simple rearrangements of each other must always have the same number of columns (or same number of scheduling steps). They have the same number of rows only if the number of facilities is equal to the number of articles.

The Gantt charts constructed above are said to be non-quantitative, because they do not take account of the actual time intervals necessary to perform each operation. They assume that $b_{ij} = 1$ for all i and j, when b_{ij} is the time to perform the operation (c_i, f_j).

5.24 Quantitative Gantt chart matrix

The Gantt chart matrices become quantitative when they take into account the actual time to perform the operation and the actual duration of the idle times. To incorporate this numeric aspect into the Gantt chart matrix, we assume, that the times are integral multiples of a certain time unit. Each index or dash in the matrix G(F) of G(C) is repeated a number of times equal to the multiple.

Example. For instance we assume in the example presented in section 5.22 that we know the actual operation time for each commodity on each facility. The times may be given within a matrix, arranged in the same way as the sequence matrix. The operation time b is given, for example, in the arrangement of the facility matrix F:

$$F = \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array} \begin{bmatrix} 2 & 1 & 2 & 4 \\ 1 & 5 & - & - \\ 2 & 1 & 3 & 5 \end{bmatrix}$$

with the B(F) matrix:

$$B(F) = \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array} \begin{bmatrix} 1 & 2 & 3 & 4 \\ 3 & 7 & - & - \\ 2 & 4 & 6 & 2 \end{bmatrix} \qquad (5.7)$$

The (ij)th element of the B(F) matrix is the operation time (set-up plus runtime) to accomplish the jth operation step of the ith commodity c_i on the facility shown in the (ij)th element of the F matrix.

A more compact and readable way is to combine both matrices F and B(F) (or C and B(C)), to one matrix FB (or CB, respectively). The elements of this matrix FB are ordered pairs, including first the facility (or commodity) index and second the operation time b.

$$FB = \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array} \begin{bmatrix} (2,1) & (1,2) & (2,3) & (4,4) \\ (1,3) & (5,7) & (\ -\ ) & (\ -\ ) \\ (2,2) & (1,4) & (3,6) & (5,2) \end{bmatrix} \tag{5.8}$$

$$CB = \begin{array}{c} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{array} \begin{bmatrix} (2,3) & (1,2) & (3,4) \\ (1,1) & (3,2) & (1,3) \\ (3,6) & (-\ -) & (-\ -) \\ (1,4) & (-\ -) & (-\ -) \\ (2,7) & (3,2) & (-\ -) \end{bmatrix} \tag{5.9}$$

This compact information is exploded into the conventional Gantt chart, by repeating the index number or the dash of FB or FC, respectively, b times as indicated by the second element of the ordered pair and applying the rules of the general Gantt chart:

$$G(FB) = \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array} \begin{bmatrix} 2 & - & - & 1 & 1 & 2 & 2 & 2 & 4 & 4 & 4 & - & - & - & - & - & - \\ 1 & 1 & 1 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & - & - & - & - & - & - & - \\ - & 2 & 2 & - & - & 1 & 1 & 1 & 1 & 3 & 3 & 3 & 3 & 3 & 3 & 5 & 5 \end{bmatrix} \tag{5.10}$$

Time Periods $\longrightarrow$

$$G(CB) = \begin{array}{c} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{array} \begin{bmatrix} 2 & 2 & 2 & 1 & 1 & 3 & 3 & 3 & 3 & - & - & - & - & - & - & - & - \\ 1 & 3 & 3 & - & - & 1 & 1 & 1 & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & 3 & 3 & 3 & 3 & 3 & 3 & - & - \\ - & - & - & - & - & - & - & - & 1 & 1 & 1 & 1 & - & - & - & - & - \\ - & - & - & 2 & 2 & 2 & 2 & 2 & 2 & 2 & - & - & - & - & - & 3 & 3 \end{bmatrix} \qquad (5.11)$$

Note, at this point, that G(F) and G(C) will always have the same number of columns if they are obtained from the same sequence matrices F and C. They will have the same number of rows only if the number of commodities is equal to the number of facilities, i.e., if $m = n$. The number of columns in the matrix G(F) or G(C) is the length of time to complete all operations in the schedule and is called the <u>length of the Gantt chart</u> matrix.

<u>Property</u>. As we know already from the Gantt chart description, each facility or machine can only process one commodity or part at a time. For the Gantt chart matrix we receive the following property: I. Each column includes not more than once a specific index number i.

To process the lot of a commodity on a facility a certain time, represented by the multiple b, is necessary. An interruption of this process time, by using the facility for another part, would require new set-ups. Thus, the operation times are only true, if the lot of the commodities are processed continuously as indicated by the operation time b. For the Gantt chart matrix this means: II. Each index has to be repeated as much in a row,

as it is indicated by the second figure in the FB or CB matrix. No space or other index is allowed within this array.

5.25 Summary

The matrix use for the Gantt chart enables the application of the advantages of the computer for the Gantt chart generation. For the storage within the computer a similar form of FB or CB will be efficient, for the printout the form of G(FB) and G(CB) gives a more readable chart.

5.3 SELECTION RULES [28], [40], [42]

To develop the scheduling method with selection rules we look at the scheduling problem from a different point of view. The machines are inspected in the shop and we observe which of the machines are productive and which of the machines are idle. Such a results may be represented with a sketch, as shown in Figure 5.4. The machines are shown by rectangles, the waiting and processed lots are described by circles. The waiting lots are drawn at the left outside of the rectangles, the in process lots are shown with a circle within the rectangle.

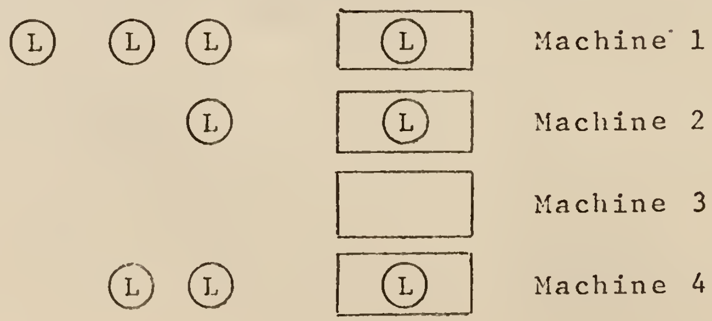


Figure 5.4. Machine loading, as a waiting line problem.

We observe, that machine 1 is productive and that there are three lots waiting to get on machine 1. Machine 2 is in production and there is only one lot waiting to be produced by machine 2. Machine 3 is idle and there is no lot available to keep this occupied. When a machine completes a lot, the foreman is confronted with the problem of scheduling. He has to make the decision: which of the waiting lots to put on the machine. In the case that there is no lot waiting, there is no decision to be made. If there is only one lot, as for machine 2, then the foreman puts this single lot on the machine. For none or one lot waiting, there is no real decision required. However, if there are several lots to be produced on a machine, then a decision has to be made as to which of these lots should be worked on first. The scheduling problem becomes one of decision making. We need a rule for the foreman whereby the foreman can make the right decision.

5.31 Simple priority

A possible decision rule might be to assign to each part and thus to each lot a priority. The lot with the highest priority is put on the machine next. This is a well defined, unambigous rule, but has a meaningful disadvantage. Some parts with a low priority might be permanently delayed. For every part is required in the manufacturing process, this means that the whole manufacturing might come to a standstill or at least delayed.

5.32 Arrival priority (first-on-first-off priority)

Another possible decision rule is to work on the lot that arrives first at the machine. This system insures that no part is delayed indefinitely, as in the first decision system. However, if a lot slows down the manufacturing process and has no real importance or priority, it can stay in the shop for a very long time and delays other lots more and more. The arrival priority requires a tight control of the detail scheduling.

5.33 Chance priority

It is possible to state a decision rule with the use of a chance variable for scheduling. Like a roulette wheel we determine by chance what part should be manufactured next. This situation leads to a somewhat better production than the previous methods. As time goes on every lot would eventually be manufactured. Such a method would tend to equalize the in-process time for different parts.

All these decision rules have an important weakness. Usually the manufacturer has to meet some shipping schedule. None of the methods are based on this fact. We require a decision rule that depends on the shipping schedule.

5.34 Delay-time priority

The decision rule, working with the delay-time priority, is based on the lot completion date. We assume, that some lots are on schedule, some are early and some will be delayed. The more a lot is delayed, the higher should be the priority. For every lot a priority based on the deviation from the standard schedule is determined.

5.4 SCHEDULING WITH MANUFACTURING BAND AND DELAY-TIME PRIORITY [38], [40], [42]

This manufacturing scheduling model incorporates two very real aspects of scheduling. The overall scheduling is determined with the aid of manufacturing bands, then the detail scheduling is made by the foreman with the help of a predetermined decision function. The decision function is based on the amount of delay-time a lot has.

5.41 Production requirements

We assume that the requirements of each part are given and have a certain distribution over future planning periods. This requirement distribution was obtained from the requirement computation (Chapter 3) and the optimal lot-size calculation (Chapter 4). We have no authority to change this requirement in any way.

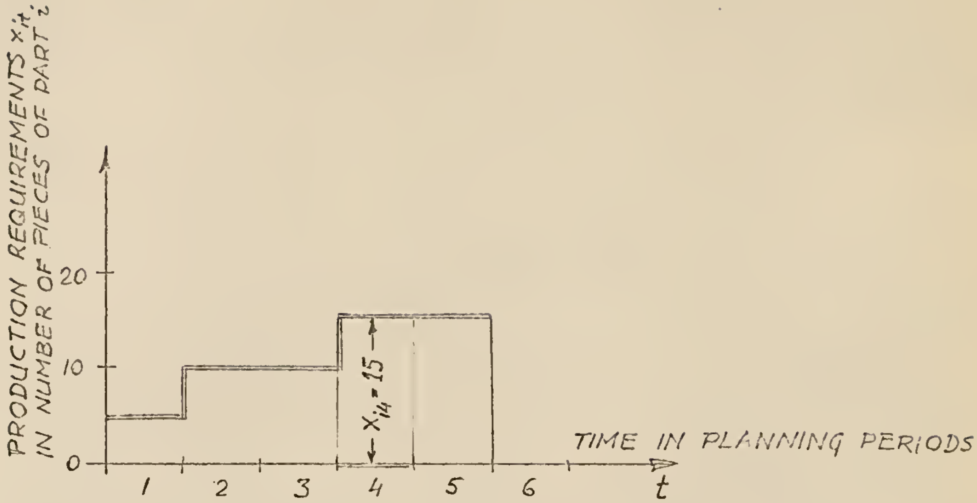


Figure 5.5. Requirement distribution of a certain part.

The requirement, usually given in table form, can be graphical represented as time dependent function (see Figure 5.5). In the previous section 4.2 we mentioned that it is much more convenient to operate with the cumulated requirement function instead of the requirement distribution. If x_{it} is the requirement of part i in the planning period t, then the cumulated number of parts in period t is computed with

$$X_{it} = \sum_{t=1}^{t} x_{it} \tag{5.12}$$

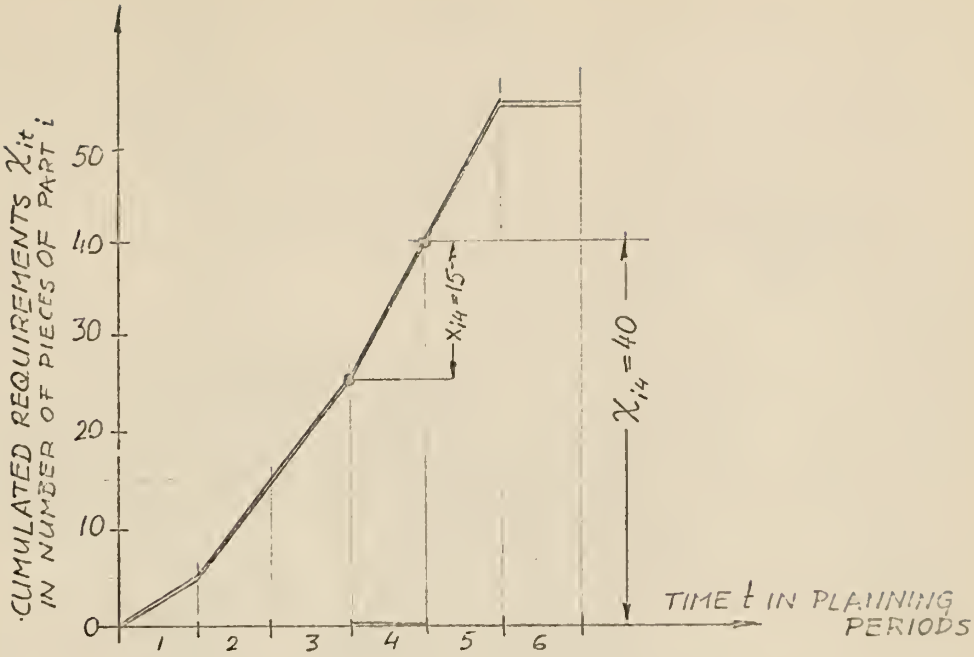


Figure 5.6. Cumulated requirement function of the same part shown in Figure 6.4.

Figure 5.6 shows the corresponding cumulated requirement function to the requirement distribution displayed in Figure 5.5.

5.42 Manufacturing bands [41]

The cumulated requirement function shows how many articles have to be finished at a specific time period. This function is our basic schedule for the whole production and usually called the shipping schedule. To satisfy the demand of our customer at the given time period, the production has to start a certain time span ahead of the completion or shipping date. Mathematically speaking, the starting schedule of a specific part can be determined by shifting the shipping or completion schedule by

the make span plus a safty cushion on the graph to the left. If S(t) defines the function of the completion schedule. The make span of part i includes set-up time as well as the effective run time.

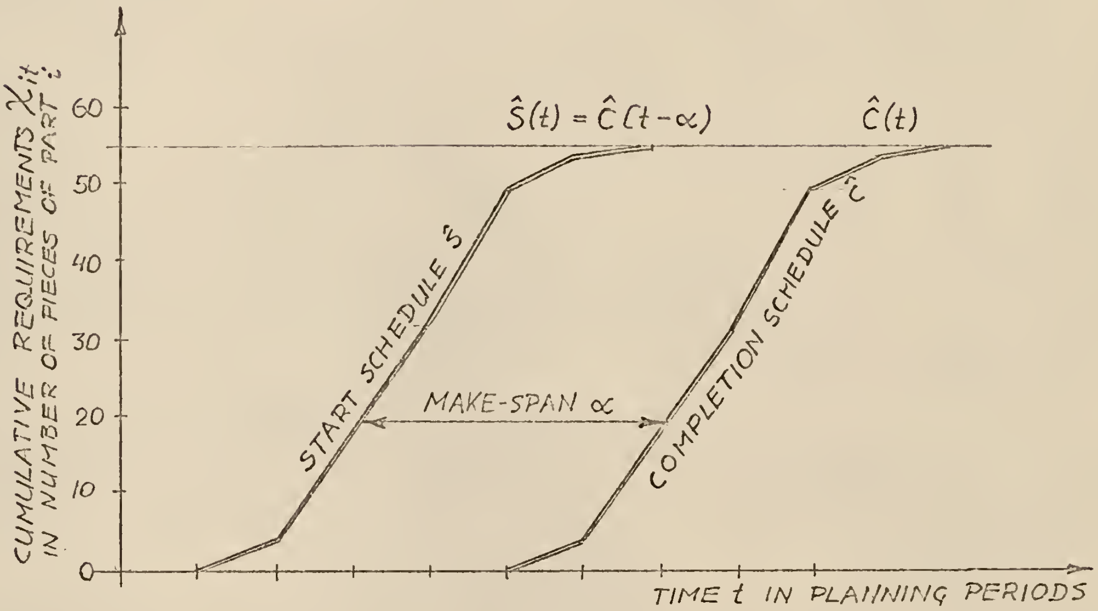


Figure 5.7. Manufacturing-band.

We call this combination of start schedule and completion schedule the manufacturing band for the specific part i. As long as the part is made on a continuous assembly line, these manufacturing bands represent the start and completion schedules. However, what do we do when the parts are manufactured in lots as we assume in the job production?

5.43 Manufacturing band and lot-production

A production part might go through many machines and there may be operations on each of these machines. The part itself can be produced in any specific lot size. The only important point is that all starting dates and all the completion dates are within the manufacturing band. Some typical examples will illustrate the use of manufacturing bands for production with lots.

Example 1. Each week we manufacture a lot of a certain part. We specify, that each of the lots has to be within the manufacturing band as shown in Figure 5.8. Each lot is to be started and completed so that the rectangle formed by the start and completion dates lies entirely inside of the manufacturing band.

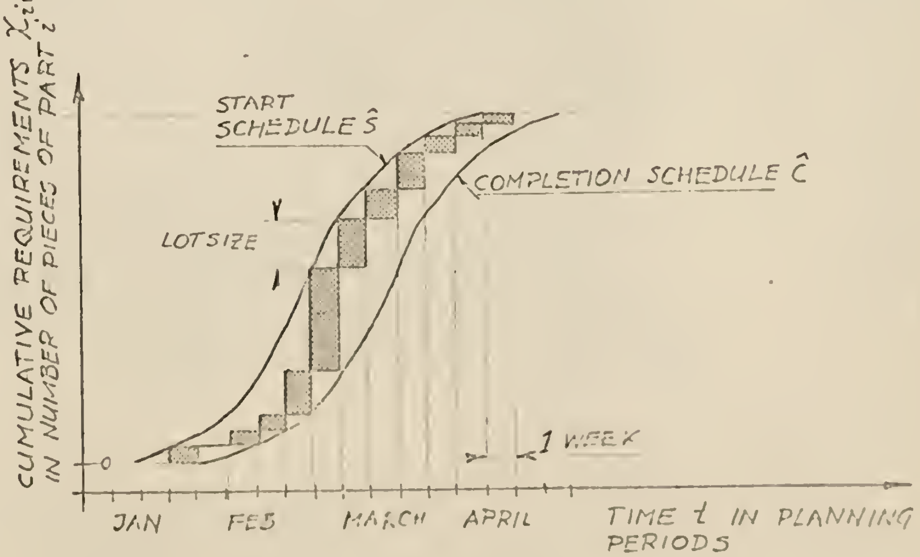


Figure 5.8. Manufacturing band with weekly release of the production lots.

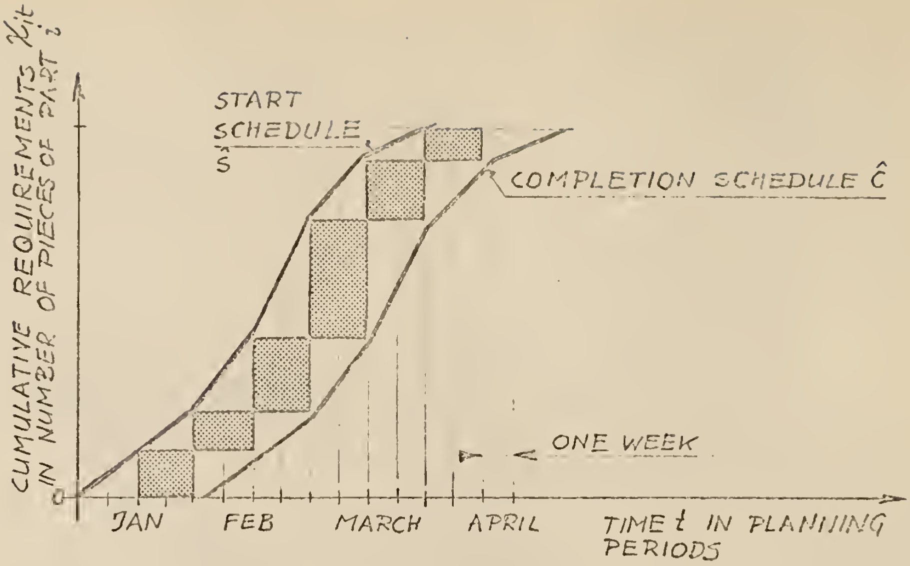


Figure 5.9. Manufacturing band with biweekly release of the production lots.

Example 2. In this example we have a variable production cycle. The part is produced in lots of a constant size. The economic lot size had been found and this is now applied on the manufacturing band scheduling. In this case the same rule is valid: the rectangle formed by start and completion date has to be entirely within the manufacturing band. Example 2 is drawn in Figure 5.10.

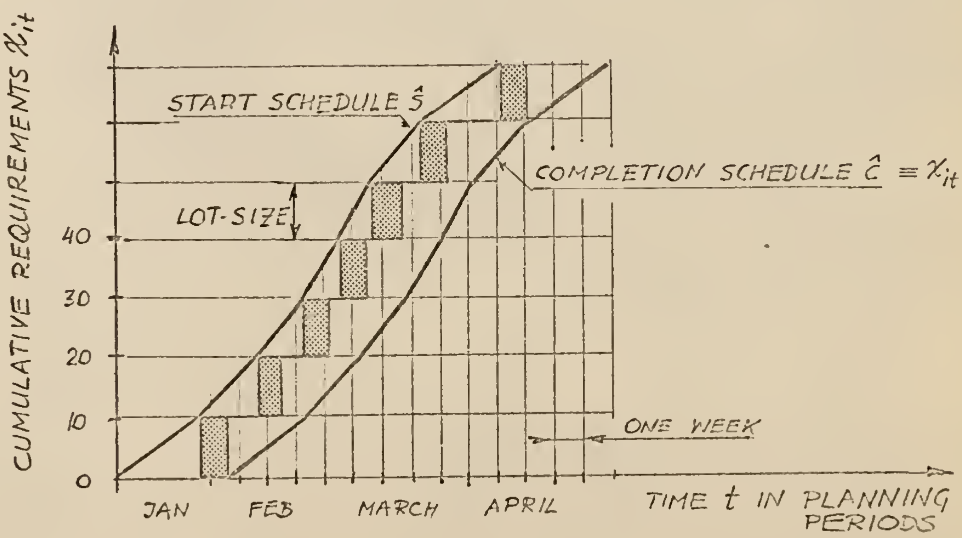


Figure 5.10. Manufacturing band with production in constant lot sizes.

Example 3. A more general interpretation of the concept of the manufacturing band is shown in Figure 5.11. A part might go through many production facilities and there might be a certain number of operations on each of the machines. The part shown in Figure 5.11 goes through four operations. The start and the completion dates are given for each of the operations. The rectangle formed by start and completion date and the lot size lies within the manufacturing band.

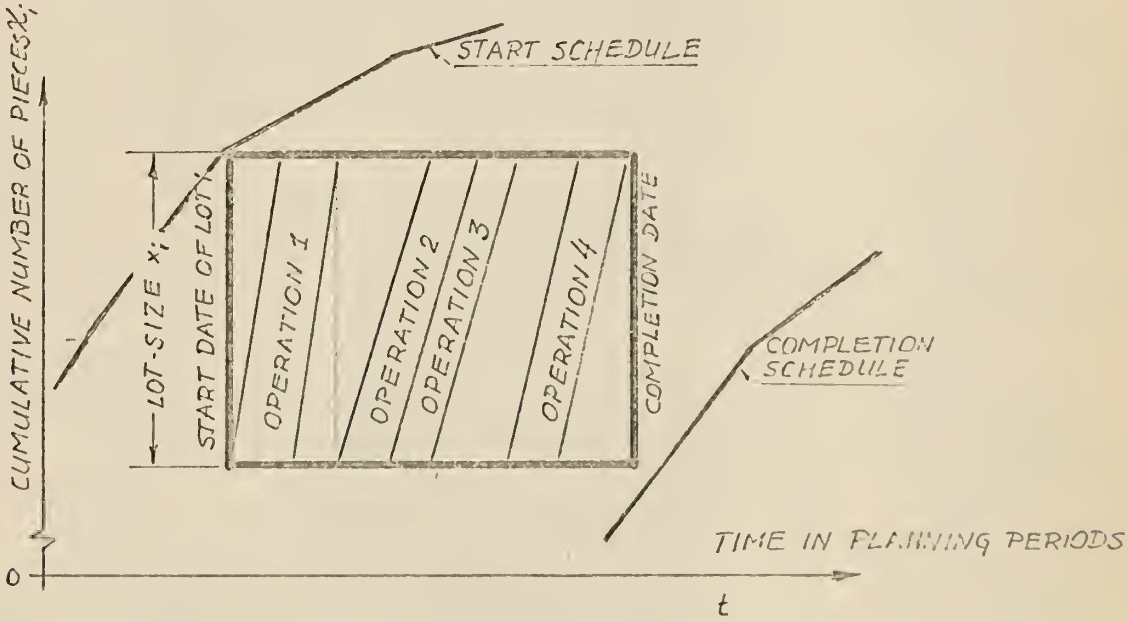


Figure 5.11. Manufacturing bands with an actual production, showing each operation.

5.44 Determination of the width of the manufacturing band

1. Set-back chart (for assembly-flow production)

To determine the width of the manufacturing band we construct the set-back chart (Figure 5.12) in accordance with the

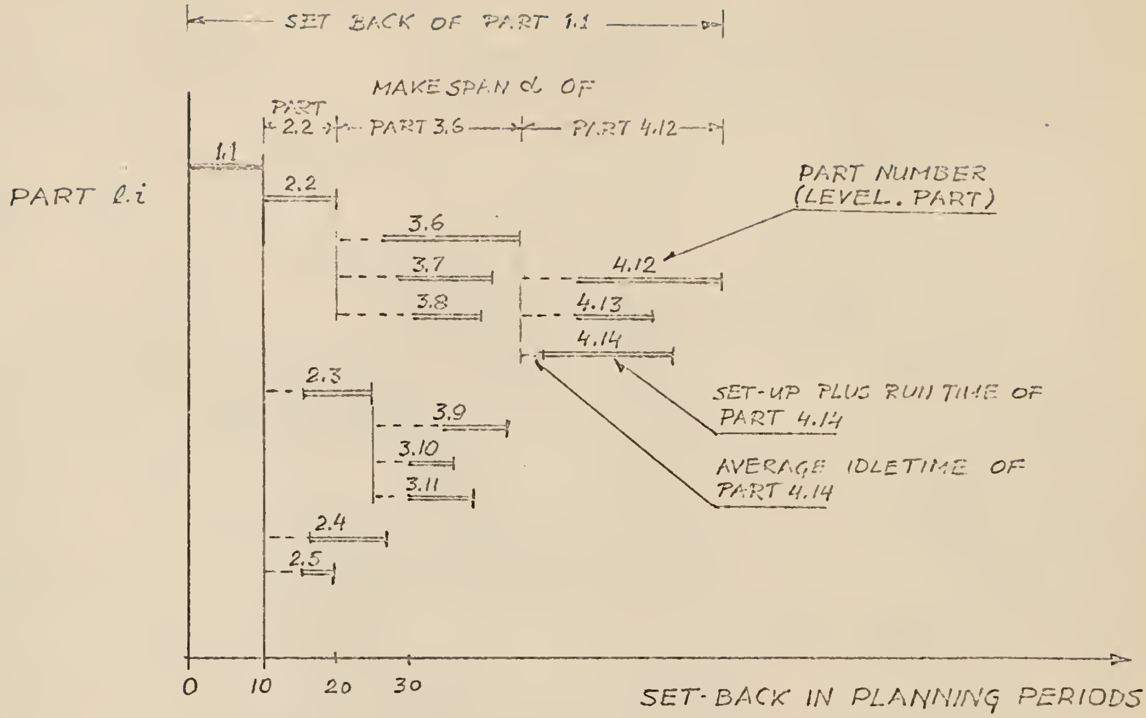


Figure 5.12. Set-back chart of part 1.1

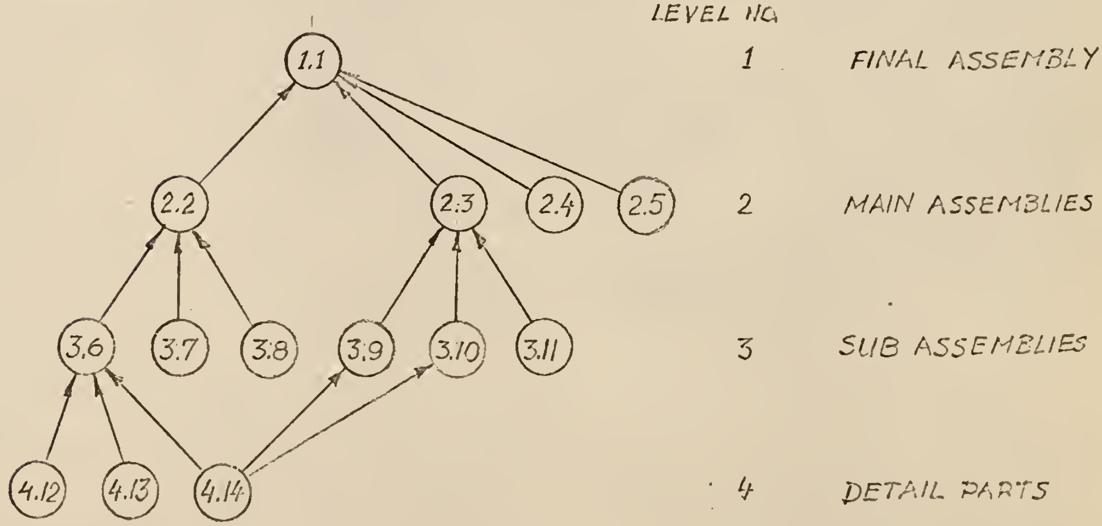


Figure 5.13. Part structure diagram of part 1.1

Part-structure graph (Figure 5.13), by allowing a fixed make span and a safety cushion for each assembly or part. As long as we deal with assembly-flow production this method is suitable. However, if we deal with parts that are made in lots, we must decide what the make span is.

2. Make span analysis (for lot production)

A manufacturing analysis of each part shows, how much set up and run time is required for each individual lot. This gives us the minimum time a part requires to be pushed through the workshop, but not the total make span. As we know, the lot may be idle on the floor waiting to be worked on thus the total makespan depends on how much time a lot has to wait as well as on the set-up and run time. The complete solution of the scheduling problem in the job-type production would allow us to calculate this idle time. However, the complete and detailed solution is unknown. If we know it we would not need to schedule with the aid of manufacturing bands.

3. Statistical approach for the make span

One approximate method is to analyze the operations in the workshop and to establish a statistical rule. This rule should give us the make span for each part.

Vazsonyi [41] set the hypothesis, that the make span of each part depends only on the number of operations involved and the total standard time required to manufacture. He obtained the following result: The make span in fact correlates well with the number of operations and depends not too much on the standard times. Figure 5.14 shows a scatter diagram of the make span and versus the number of operations ω to be performed on each part.

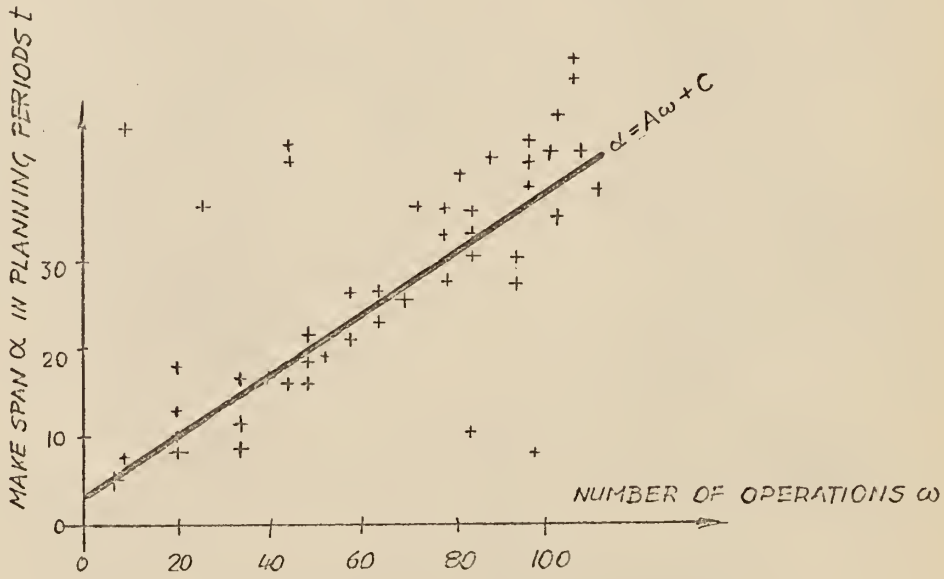


Figure 5.14. Relationship between the number of operations ω and the make span α.

Dealing with a production, whose make span is approximately independent of the Labor hours required to make each part, we state:

$$\alpha = A\omega + C \tag{5.13}$$

Under more general conditions we have to assume, that the make span α depends on labor hours required. Denoting the labor hours to make a particular part with τ, we enlarge the equation (5.13) to:

$$\alpha = A\omega + B \cdot x_{lot} \cdot \tau_{\omega} + C \qquad (5.14)$$

where the symbols have the following meaning:

α = make span in production periods,

ω = total number of operations,

τ_{ω} = total standard time to produce one single part on the ω operations.

A = constant,

B = constant,

C = constant,

x_{lot} = lot size

The last equation (5.14) is more general than equation (5.13) and has a wider application. The method of correlating the make span with the number of operations is only valid for the normal range of operations and lot sizes observed in the study. It is not advisable to extrapolate.

5.45 Decision making with delay-time priority

We have specified the manufacturing bands, but we did not mention how the detail scheduling is performed. As long as the actual production stays within the manufacturing bands the shipping schedule will be met. The width of the manufacturing bands make it difficult to survey the progress of the production, in relation to the completion date.

5.45.1 Absolute delay time

To obtain a certain guide within the manufacturing band we develop a tool which indicates the progress or the delay of each lot in the production. We want to know where we are with the actual production within the manufacturing band. An economical way to obtain this goal is to compute the progress or the delay of each lot every day. The lot with the largest delay needs processing more urgently than any other and should be preferred to all the other lots. The "absolute delay time Δt" is the difference between the actual date t_{actual} and the theoretical date t_{theor}. The theoretical date indicates the time, where the production should be with the already completed work. Defining

$\hat{S}$ - as the indate or start date of the manufacturing band,

n - as the number of completed operations, and

τ_n - as the total standard time to produce one single part up to the nth operation,

we calculate with linear interpolation the time t_{theor}

$$t_{theor} = \hat{S} + (An + Bx_{lot}\tau_n + C) \tag{5.15}$$

and the absolute delay time Δt with

$$\Delta t = t_{actual} - t_{theor} = t_{actual} - \hat{S} - (An + Bx_{lot}\tau_n + C) \tag{5.16}$$

5.45.2 Relative delay

To compare the delay, the absolute delay time $\Delta t = t_{actual} - t_{theor}$ is not sufficient. This delay time has no base to compare lots with different width manufacturing bands. A ratio of this absolute delay and the total available make span gives a sound base to compare and to decide upon. The relative delay is computed with the fraction

$$\lambda = \frac{\text{absolute delay}}{\text{make span}} = \frac{t_{actual} - t_{theor}}{\hat{S} - \hat{C}} = \frac{\Delta t}{\alpha} \tag{5.17}$$

where:

t_{actual} = the current date

t_{theor} = the date where the lot should be with the completed operations n

$\hat{S}$ = the start date or in-date

$\hat{C}$ = the completion date or out-date

α = the make span

This ratio λ is called "relative delay" and can be used as a decision parameter. The bigger the delay is relative to the make span the more urgent this part becomes. We express this statement with the decision parameter and state: The lot with the largest decision parameter has the highest priority. It has to be taken first out of the waiting line, for processing.

In practice the scheduling model works as follows: Every time a lot is completed on a machine, the foreman surveys the lots that are waiting, computes the decision parameter and selects the lot with the largest decision parameter λ.

5.45.3 Decision function

With the decision parameter λ we have a scheduling model, that removes decision making from the foreman. Using the models mentioned in section 5.3 the foreman had too many decisions to perform, almost without any objective guide; now he has no decision freedom at all. To make the model more flexible and to give the foreman a certain responsibility with decision making, we introduce a function that transforms the decision parameter λ into another decision parameter δ. One obvious very suitable function is $\delta = \delta(\lambda)$, which is defined as following:

$$\delta = \begin{cases} -1 & \forall \quad \lambda \leq -1 \\ \lambda & \forall \quad \lambda > -1 \end{cases}$$

The picture of this function 0 is shown in Figure 5.16.

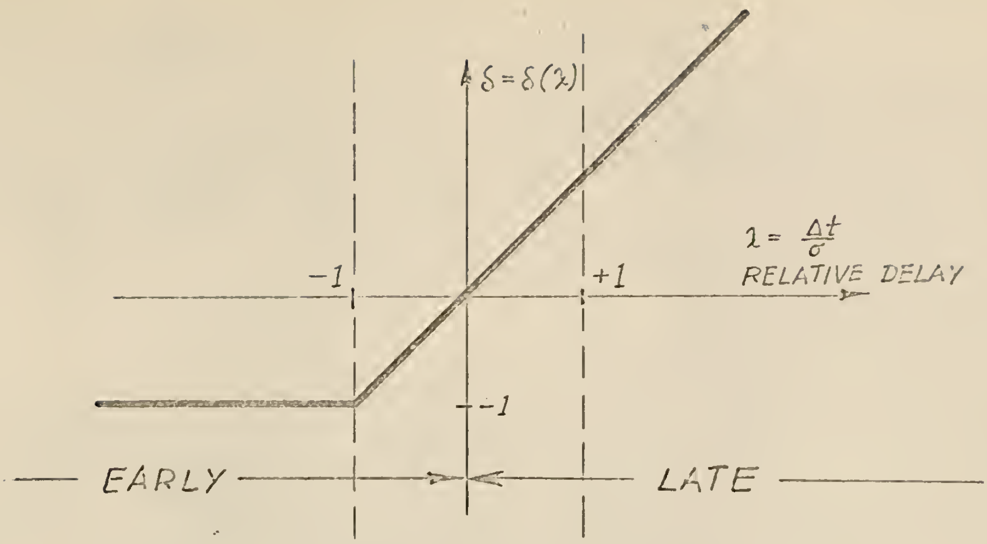


Figure 5.16. Decision parameter λ and δ.

Here, similar to the original decision parameter, the larger the decision parameter, the more priority is given to the individual lot. With this parameter δ the foreman has no choice for late parts, but has more flexibility for early parts.

The decision function could have different shapes, depending on the nature of the problem. The important point is, that with the aid of a decision function, preciseness and flexibility can be introduced into the scheduling model.

5.45.4 Example

Let us consider that the foreman is confronted with the problem of deciding which of the seven lots, shown in Table 5.15, should be put on the machine next. All seven lots are waiting before a certain machine to be processed.

The Table 5.15 indicates the progress of these lots in the workshop. Beginning with the first column the table shows:

the lot number,

the total number of operations necessary to finish the parts,

the make span σ, which was calculated with equation (5.13)

$\alpha = A\omega + C$, using $A = 2$ and $C = 5$,

the start date S,

the completion date C, and

the number of already completed operations n.

We investigate which of the lots should be taken first? The number of completed operations n is also enclosed in the table, as reported from the workshop. For the first lot 20 operations have been completed. The start date for the first lot is S = 195. The manufacturing day on which the first n operations have to be completed is computed with

$$t_{theor} = S + (A.n + C) \tag{5.18}$$

and for our first lot:

$$t_{theor} = 195 + (2.20 + 5) = 240$$

We see that on the manufacturing day 240 the first 20 operations should have been completed, but the current manufacturing day is $t_{actual} = 250$. This shows, that this lot is late by $\Delta t = 250 - 240 = 10$ manufacturing days.

Lot number	Total Number of operations	Make span	Start date	Completion date	Completed operations		Absolute delay	Relative delay	Decision function
	ω	α	$\hat{S}$	$\hat{C}$	n	t_{theor}	Δt	λ	δ
		$\alpha = A\omega + c$	$\hat{S} = \hat{C} - \alpha$	given	report from the workshop	$t_{theor} = S + (\Lambda . n + c)$	$\Delta t = t_{actual} - t_{theor}$	$\lambda = \frac{\Delta t}{\alpha}$	$\delta = \begin{cases} -1 & \forall \lambda \leq -1 \\ \lambda & \forall \lambda > -1 \end{cases}$
--	no.	days	days	days	no.	days	days	--	--
1	50	105	195	300	20	240	10	0.095	0.095
2	60	125	185	310	25	240	10	0.080	0.080
3	100	205	210	415	10	235	15	0.066	0.066
4	20	45	246	291	7	265	-15	-0.79	-0.79
5	25	55	277	332	19	320	-70	-1.27	-1.00
6	10	25	195	220	5	210	40	1.6	1.60
7	15	35	321	356	2	330	-80	-2.3	-1.00

Table 5.15. Workshop scheduling with the aid of "decision function δ". Current manufacturing day $t_{actual} = 250$.

5.45.5 Control of the production situation with the decision parameter

Collecting all the decision parameters and plotting them in a frequency diagram we obtain graphs as shown in Figure 5.17. If most of the decision parameters are positive, the distribution looks like the right graph in Figure 5.17. Most of the parts are late and the production as a whole is late. On the other side, if we obtain a curve similar to the left side of Figure 5.17, most of the parts are ahead the schedule and thus the shop as a whole is ahead.

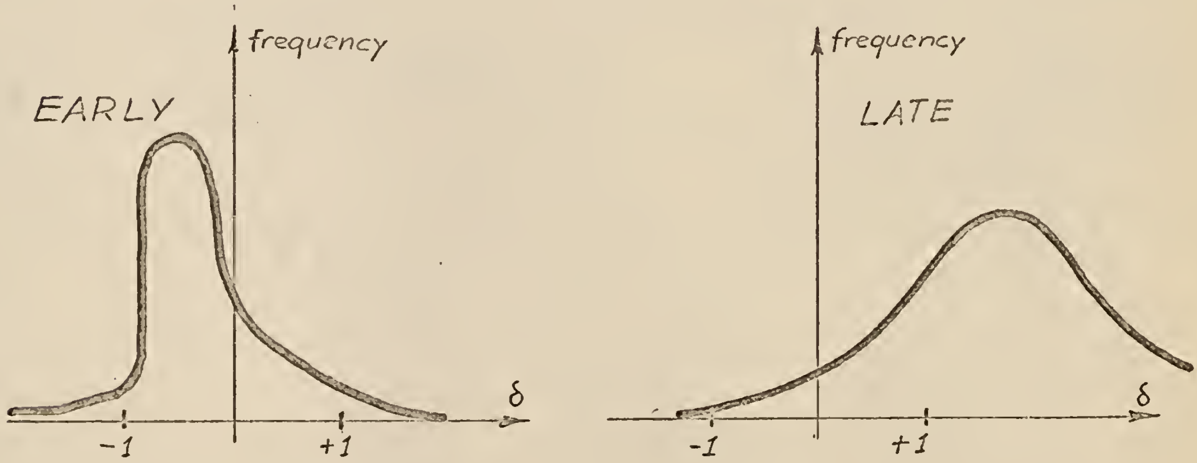


Figure 5.17. Distribution of the decision parameter.

If the variability of the decision parameter is small, we obtain a narrow distribution curve. This indicates the schedule is tightly controlled. If the curve is spread out, the parts are loosely controlled. See Figure 5.18.

Summarizing, the distribution of the decision parameter λ or δ represents an objective quantitative measure of the earliness or lateness (with the location of the mean value of the parameter) and the tightness of control (with the standard deviation of the parameter) for the production control.

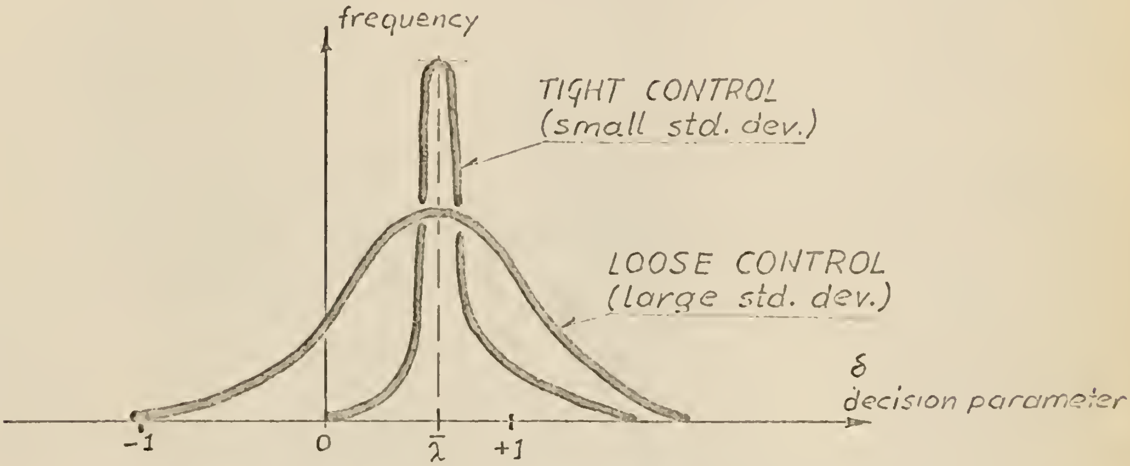


Figure 5.18. Distribution of the decision parameter.

5.5 SUMMARY

The total model. The scheduling model developed in this chapter divided decision making into two different levels. The basic manufacturing band system established an overall scheduling interval but left the detail scheduling open. The detail scheduling is performed with the aid of the decision procedures and is accomplished by a lower level of supervision.

This scheduling model combines, therefore, two practical features of the scheduling problem. The overall scheduling is executed by production control with the aid of manufacturing bands. The daily detail scheduling is performed by the foreman in the shop with the help of decision procedures.

6. CONCLUSION

This thesis shows one way to develop economically a feasible schedule from a given sales forecast. The method is applicable for job and lot production, where a number of identical articles are produced. A refined matrix algebra and a wise arrangement of the part hierarchy permit the efficient computation of the problem. For the immediate scheduling work a two part method is suggested: The over-all scheduling is done into manufacturing time bands, and the detail-scheduling within the bands by loading rules for the facilities. The loading algorithm has already been applied in industry and has recorded acceptable results.

Future investigations should consider the following aspects:

1. Stating the requirements problem with a realistic continuous time scale rather than discrete periods,
2. A better integration of requirements with optimal lot size and scheduling,
3. Application of the network concepts such as PERT to the requirement and the scheduling routine,
4. Simplifying and reducing the computations and the computation time, and
5. Transformation of these algorithms into a workable computer program.

7. REFERENCES

1. S. B. Akers, Jr. and J. Friedman, "A non-numerical approach to production scheduling problems", Operation Research 3, 1955, 429-422.

2. C. T. Baker, "Fabrication and assembly operations. Part I: The outlines of a control system." IBM Systems Journal, Vol. 4, No. 2, 1965, 86-93.

3. A. B. Calica, "Fabrication and assembly operations. Part II: Long-range planning techniques", IBM Journal, Vol. 4, No. 2, 1965, 94-104.

4. F. L.Church, "Requirements generation, explosions and bills of material", IBM System Journal, September-December 1963, 268-287.

5. R. W. Conway, W. L. Maxwell, L. W. Miller, "Theory of scheduling", Addison Wesley Publishing Company, Reading, Massachusetts, 1967.

6. C. R. Decarlo, "The use of automatic and semi-automatic processing equipment in production and inventory control", In the conference on operation research in production and inventory control, Case institute of Technology, 1954.

7. B. P. Dzielinski, "Fabrication and assembly operations. Part IV: Linear programming in production planning", IBM Systems Journal, Vol. 4, No. 2, 1965, 122-135.

8. B. P. Dzielinski, C. T. Baker and A. S. Manne, "Simulation tests of lot size programming", Management Science, Vol. 9 No. 2, Jan. 1963, 229-257.

9. S. Eilon, "Elements of production planning and control", The Macmillan Company, New York, 1962.

10. S. E. Elmaghraby, "A note on the explosion and netting problems in the planning of materials requirements", Operation Research, Vol. 11, No. 4, July-August 1963, 530-535.

11. S. E. Elmaghraby, "An algebra for the analysis of generalized activity networks", Management Science, Vol. 10, No. 3, April 1964, 494-514.

12. B. Giffler, "Mathematical solution of production planning and scheduling problems", Doct. Dissert., Columbia University, School of Engineering, 1960.

13. B. Giffler, "Mathematical solution of production planning and scheduling problems", IBM Technical Report 09.01028.026, 2nd printing, November 1961.

14. B. Giffler, "Scheduling general production systems using scheduling algebra", Naval Research Logistics Quarterly, Sept 1963, 237-255.

15. B. Giffler, "Schedule algebras and their use in formulating general systems simulations", Industrial Scheduling, edited by John Muth and Gerald Thompson, Prentice Hall, 1963, 39-59.

16. B. Giffler, "Simulation models in production scheduling and inventory control", Production & Inventory Management, July 1966, 1-14.

17. B. Giffler and C. L. Thompson, "Algorithms for solving production scheduling problems", Operation Research, July 1960, 487-503.

18. B. Giffler, G. L. Thompson and V. Van Ness, "Numerical experience with the linear and Monte Carlo algorithms for solving production scheduling problems", 1960.

19. L. Gleiberman, "The engineering change of the total·requirement matrix for a bill of materials", Management Science 10, No. 3, April 1964, 488-493.

20. C. C. Holt and H. A. Simon, "Optimal decision rules for production and inventory control" Operation research in production and inventory control, Case Institute of Technology 1954, 73-89.

21. E. D. Homer, "Some remarks an the computation of production requirements", paper presented at the 1962 International Meeting of the Institute of Management Sciences, University of Michigan, Ann Arbor, Michigan, Sept. 1962.

22. R. Jackson, "Notes on some scheduling problems", Management Sciences Res. Project, Research Report No. 35, UCLA, Oct. 1, 1954.

23. S. M. Johnson, "Optimal two and three-stage production schedules with set-up times includes", The Rand Corporation, P-402, May 1953; Industrial Scheduling by John Muth and Gerald Thompson, Prentice Hall, 1963, 13-20.

24. J. E. Kelley, "Critical-path planning and scheduling: mathematical basis", Operation Research, Vol. 2 , May-June 1961, 296-320.

25. R. S. Ledley, "Mathematical foundation and computational methods for digital logic machine", Operation Research, 1954, 249-274.

26. I. J. Lieberman, "A mathematical model for integrated business systems", Management Science, Vol. 2, No. 4, July 1956, 327-336.

27. P. G. Loewner, "Fabrication and assembly operations. Part III; Matrix methods for processing configuration data", IBM System Journal, Vol. 4, No. 2, 1965, 105-121.

28. A. S. Manne, "Programming of economic lot sizes", Management Science, Vol. 4, No. 2, January 1958, 115-135.

29. J. A. Parsons, "Multiproduct lot size determination when certain restrictions are active", The Journal of Industrial Engineering, Vol. 18, No. 7, July 1966, 360-365.

30. M. E. Salveson, "On a quantitative method in production planning and scheduling", Econometrica, October 1951, 554-590.

31. M. E. Salveson, "Mathematical methods in management programming", Case Institute of Technology 1954, Operation Research in Production and Inventory Control, 23-41.

32. P. P. Schoderbek and L. A. Digman, "Third generation, RERT/LOB", Harvard Business Review, September-October 1967, 100-110.

33. H. A. Simon, "On the application of servomechanism theory in the study of production control", Econometrica, Vol. 20, April 1952, 247-268.

34. B. C. Spradlin and D. A. Pierce, "Production scheduling under a learning effect by dynamic programming", The Journal of Industrial Engineering, Vol. 18, No. 3, March 1967, 219-222.

35. M. K. Starr and D. W. Miller, "Inventory control: theory and practice", Prentice Hall, Inc., 1962, Chapter 4, 8 and 9.

36. A. W. Steinberg, "Some notes on the similarity of three management science models and their analysis by connectivity matrix techniques", Management Science, Vol. 9, No. 2, January 1963, 341-343.

37. A. Vazsonyi, "Production control from the point of view of decision theory", (Abstract), Management Science 1, 190, January 1955.

38. A. Vazsonyi, "The use of mathematics in production and inventory control", Management Science 1, No. 1, Oct. 1954, 70-85 and No. 3 , April-July 1955, 207-223.

39. A. Vazsonyi and H. T. Larson, "Data-processor requirements in production and inventory control", Proceedings of the Western Joint Computer Conference, 1955, 48-61.

40. A. Vazsonyi, "Operation reserach in production control - a progress report", Journal of the Operation Research Society of America, Vol. 4, 1956, 19-31.

41. A. Vazsonyi, "Economic-lot size formulas in manufacturing", Operation Research, Vol. 5, No. 1, Feb. 1957, 28-44.

42. A. Vazsonyi, "Scientific programming in business and industry", John Wiley & Sons, 1958, Chp. 13, 429-467.

43. F. V. Waugh, "Inversion of the leontief matrix by power series", Econometrica, April 1950, 142-154.

44. H. H. Young, "Optimization models for production lines", Journal of Industrial Engineering, Vol. 18, No. 1, January 1967, 70-78.

MATHEMATICAL FORMULATION OF JOB SHOP SCHEDULING

by

FRITZ NENTWICH

I. Staatspruefung, University of Technology,
Vienna, Austria, 1963

Dipl. Ing., University of Technology,
Vienna, Austria, 1966

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1968

This thesis describes the production planning of requirements and scheduling for job type production. The assumed production model consists of a sequence of operations that manufacture and assemble several components to one assembly on a certain number of facilities.

The efficient computation of the time-dependent requirement uses a refined matrix algebra combined with an efficient part arrangement in an assembly hierarchy. This requirement problem incorporates the description of part structure and storage; part explosion; reduction of requirements by inventory; determination of lot size; and manufacturing time span.

Scheduling itself is performed at two levels: Rough scheduling of components and assemblies to "manufacturing time-bands" and detailed scheduling of components and assemblies within this band. An economical loading rule for detailed scheduling is described.